



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**FEASIBILITY STUDY OF VOIP INTEGRATION INTO
THE MYSEA ENVIRONMENT**

by

Lily Tse

September 2005

Thesis Advisor:

Co-advisor:

Cynthia E. Irvine

Thuy D. Nguyen

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE September 2005	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE: Feasibility Study of VoIP Integration into the MYSEA Environment			5. FUNDING NUMBERS	
6. AUTHOR(S) Lily Tse				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) <p>Voice over Internet Protocol (VoIP) is becoming popular due to its low cost and the management advantages it offers over traditional PSTN phone systems. VoIP is widely implemented with H.323 and Session Initiation Protocol (SIP) standards. However, both protocols are poorly designed for networks with common security solutions such as firewalls and Network Address Translation (NAT).</p> <p>This project is a feasibility study of SIP-based VoIP integration into the Monterey Security Architecture (MYSEA), a multilevel secure environment that uses NAT as a security mechanism. A gathering of comparative studies on VoIP protocols was performed to guide the selection of SIP as the test protocol. A set of experiments was devised and conducted using SIP-based softphones for this study. The insights gained from the experiment provide useful insights to the MYSEA project concerning VoIP security.</p>				
14. SUBJECT TERMS Voice over Internet Protocol, H.323, Session Initiation Protocol, Network Address Translation, Monterey Security Architecture			15. NUMBER OF PAGES 204	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**FEASIBILITY STUDY OF VOIP INTEGRATION INTO THE MYSEA
ENVIRONMENT**

Lily Tse
Civilian, Naval Postgraduate School
B.S., University of California, Davis, 2003

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

**NAVAL POSTGRADUATE SCHOOL
September 2005**

Author: Lily Tse

Approved by: Cynthia E. Irvine
Thesis Advisor

Thuy D. Nguyen
Co-Advisor

Peter J. Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Voice over Internet Protocol (VoIP) is becoming popular due to its low cost and the management advantages it offers over traditional PSTN phone systems. VoIP is widely implemented with H.323 and Session Initiation Protocol (SIP) standards. However, both protocols are poorly designed for networks with common security solutions such as firewalls and Network Address Translation (NAT).

This project is a feasibility study of SIP-based VoIP integration into the Monterey Security Architecture (MYSEA), a multilevel secure environment that uses NAT as a security mechanism. A gathering of comparative studies on VoIP protocols was performed to guide the selection of SIP as the test protocol. A set of experiments was devised and conducted using SIP-based softphones for this study. The insights gained from the experiments provide useful insights to the MYSEA project concerning VoIP security.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	MOTIVATION	1
B.	PURPOSE OF STUDY.....	1
C.	ORGANIZATION OF PAPER	1
II.	BACKGROUND	3
A.	INTRODUCTION.....	3
	1. Advantages of VoIP	3
	<i>a. Efficient Use of Bandwidth.....</i>	<i>3</i>
	<i>b. Reduction or Possibly Elimination of Long Distance and Phone Charges</i>	<i>4</i>
	<i>c. Convergence of Voice and Data Networks.....</i>	<i>4</i>
	<i>d. Advanced Features.....</i>	<i>4</i>
	2. Disadvantages of VoIP.....	4
	<i>a. Quality of Voice.....</i>	<i>4</i>
	<i>b. Security.....</i>	<i>5</i>
	<i>c. Availability.....</i>	<i>5</i>
	<i>d. 911.....</i>	<i>5</i>
B.	VOIP OVERVIEW	5
	1. VoIP Phone Overview.....	6
C.	VOIP SERVICES.....	7
	1. Encoding	8
	2. Transport.....	8
	3. Gateway Control	8
D.	VOIP PROTOCOLS	8
	1. H.323	9
	2. Session Initiation Protocol.....	9
	3. MGCP	9
	4. Megaco/H.248.....	10
E.	VOIP CHALLENGES.....	10
	1. Quality of Service.....	10
	2. Security	10
F.	NAT	12
	1. netfilter/iptables	13
	2. SIP with NAT	13
G.	MYSEA OVERVIEW	14
H.	SUMMARY	15
III.	TECHNICAL COMPARISON BETWEEN H.323 AND SIP	17
A.	H.323 AND SIP OVERVIEW	17
	1. Background	17
	2. Architecture.....	18

3.	Components	19
4.	Call Setup.....	20
a.	H.323	20
b.	SIP	21
5.	Services.....	22
B.	H.323 AND SIP COMPARISON	23
1.	Simplicity and Flexibility	24
a.	Protocol Specification	24
b.	Message Encoding	24
c.	Protocol Interactions.....	24
d.	Applications.....	25
2.	Extensibility	25
a.	Extensible Mechanisms	25
b.	Backward-Compatibility	26
c.	Interoperability.....	26
3.	Scalability.....	26
a.	Protocol Design	26
b.	Servers	27
c.	Conferencing.....	27
4.	Security	27
a.	H.323 Security - H.235	27
b.	SIP Security.....	28
c.	Security Comparison.....	28
5.	Conclusion	30
C.	SUMMARY	30
IV.	TESTING.....	31
A.	TEST METHODOLOGY	31
B.	TEST DESCRIPTION.....	32
1.	Test 1: No NAT VoIP Configuration	32
2.	Test 2: Single NAT VoIP Configuration	32
3.	Test 3: Double NAT VoIP Configuration	34
4.	Test 4: Extended Double NAT VoIP Configuration.....	35
5.	Test 5: Extended Double NAT VoIP Configuration with Simultaneous VoIP Sessions.....	37
6.	Test 6: MYSEA Configuration	39
C.	PROBLEMS ENCOUNTERED	40
D.	TEST RESULT	41
E.	SUMMARY	41
V.	FUTURE WORK AND CONCLUSIONS	43
A.	FUTURE WORK	43
1.	Routing in MLS Server	43
2.	VoIP Conversations Initiated from the Internet.....	43
B.	CONCLUSIONS	44
	APPENDIX A. A SURVEY OF VOIP HARDPHONES AND SOFTPHONES	45

A.	HARDPHONES	45
B.	SOFTPHONES.....	46
APPENDIX B. TEST 1: NO NAT VOIP DEMONSTRATION USING SJPHONE		47
APPENDIX C. TEST 2: SINGLE NAT VOIP DEMONSTRATION USING SJPHONE		53
APPENDIX D. TEST 3: DOUBLE NAT VOIP DEMONSTRATION USING SJPHONE		69
APPENDIX E. TEST 4: EXTENDED DOUBLE NAT VOIP DEMONSTRATION USING SJPHONE		83
APPENDIX F. TEST 5: EXTENDED DOUBLE NAT VOIP WITH SIMULTANEOUS VOIP SESSIONS DEMONSTRATION USING SJPHONE		105
APPENDIX G. TEST 6: MYSEA VOIP CONFIGURATION.....		129
LIST OF REFERENCES		179
INITIAL DISTRIBUTION LIST		181

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1.	MYSEA Network Architecture [From Ref. 8]	15
Figure 2.	H.323 Protocol Stack [From Ref. 9]	18
Figure 3.	SIP Protocol Stack [From Ref. 9]	19
Figure 4.	Simple H.323 Call Setup.....	21
Figure 5.	Simple SIP Call Setup.....	22
Figure 6.	Test 1: Physical and Logical Network Topology	32
Figure 7.	Test 2: Physical Network Topology (with and without firewall)	33
Figure 8.	Test 2: Logical Network Topology (with firewall).....	34
Figure 9.	Test 3: Physical Network Topology	35
Figure 10.	Test 3: Logical Network Topology	35
Figure 11.	Test 4: Physical Network Topology	36
Figure 12.	Test 4: Logical Network Topology.....	37
Figure 13.	Test 5: Physical Network Topology	38
Figure 14.	Test 5: Logical Network Topology.....	38
Figure 15.	Test 6: Physical Network Topology	39
Figure 16.	Test 6: Logical Network Topology.....	40
Figure 17.	Test 1: Packet Capture on Client A.....	49
Figure 18.	Test 1: Packet Capture on Client B.....	50
Figure 19.	Test 2: Packet Capture on Client A (without firewall)	58
Figure 20.	Test 2: Packet Capture on Client B (without firewall)	59
Figure 21.	Test 2: Packet Capture on eth0 of NAT (without firewall)	60
Figure 22.	Test 2: Packet Capture on eth1 of NAT (without firewall)	61
Figure 23.	Test 2: Packet Capture on Client A (with firewall)	64
Figure 24.	Test 2: Packet Capture on Client B (with firewall).....	65
Figure 25.	Test 2: Packet Capture on eth0 of NAT (with firewall).....	66
Figure 26.	Test 2: Packet Capture on eth1 of NAT (with firewall).....	67
Figure 27.	Test 3: Packet Capture on Client A.....	75
Figure 28.	Test 3: Packet Capture on Client B.....	76
Figure 29.	Test 3: Packet Capture on eth0 of NAT 1	77
Figure 30.	Test 3: Packet Capture on eth1 of NAT 1	78
Figure 31.	Test 3: Packet Capture on eth0 of NAT 2.....	79
Figure 32.	Test 3: Packet Capture on eth1 of NAT 2.....	80
Figure 33.	Test 4: Packet Capture on Client A (Client B Calls Client A).....	90
Figure 34.	Test 4: Packet Capture on Client B (Client B Calls Client A).....	91
Figure 35.	Test 4: Packet Capture on eth1 of NAT 1 (Client B Calls Client A).....	93
Figure 36.	Test 4: Packet Capture on eth0 of NAT 2 (Client B Calls Client A).....	94
Figure 37.	Test 4: Packet Capture on eth1 of NAT 2 (Client B Calls Client A).....	95
Figure 38.	Test 4: Packet Capture on Client A (Client C Calls Client A).....	97
Figure 39.	Test 4: Packet Capture on Client C (Client C Calls Client A).....	98
Figure 40.	Test 4: Packet Capture on eth0 of NAT 1 (Client C Calls Client A).....	99
Figure 41.	Test 4: Packet Capture on eth1 of NAT 1 (Client C Calls Client A).....	100

Figure 42.	Test 4: Packet Capture on eth0 of NAT 3 (Client C Calls Client A).....	101
Figure 43.	Test 4: Packet Capture on eth1 of NAT 3 (Client C Calls Client A).....	102
Figure 44.	Test 5: Packet Capture on Client A.....	114
Figure 45.	Test 5: Packet Capture on Client C.....	115
Figure 46.	Test 5: Packet Capture on eth0 of NAT 2.....	116
Figure 47.	Test 5: Packet Capture on eth1 of NAT 2.....	117
Figure 48.	Test 5: Packet Capture on eth0 of Router	118
Figure 49.	Test 5: Packet Capture on eth1 of Router	119
Figure 50.	Test 5: Packet Capture on eth2 of Router	120
Figure 51.	Test 5: Packet Capture on Client B.....	121
Figure 52.	Test 5: Packet Capture on Client D.....	122
Figure 53.	Test 5: Packet Capture on eth0 of NAT 3.....	123
Figure 54.	Test 5: Packet Capture on eth1 of NAT 3.....	124
Figure 55.	Test 5: Packet Capture on eth0 of NAT 1.....	125
Figure 56.	Test 5: Packet Capture on eth1 of NAT 1.....	126
Figure 57.	Test 6: Scenario 1 Packet Capture on Router (pinged from NAT 2), Part 1 .	137
Figure 58.	Test 6: Scenario 1 Packet Capture on Router (pinged from NAT 2), Part 2 .	138
Figure 59.	Test 6: Scenario 1 Packet Capture on NAT 1 (pinged from NAT 2)	139
Figure 60.	Test 6: Scenario 1 Packet Capture on Router (pinged from Router), Part 1..	141
Figure 61.	Test 6: Scenario 1 Packet Capture on Router (pinged from Router), Part 2..	142
Figure 62.	Test 6: Scenario 1 Packet Capture on NAT 1 (pinged from Router).....	143
Figure 63.	Test 6: Scenario 2 Packet Capture on Router (pinged from NAT 2).....	148
Figure 64.	Test 6: Scenario 2 Packet Capture on NAT 1 (pinged from NAT 2), Part 1 .	149
Figure 65.	Test 6: Scenario 2 Packet Capture on NAT 1 (pinged from NAT 2), Part 2 .	150
Figure 66.	Test 6: Scenario 2 Packet Capture on NAT 1 (pinged from NAT 2), Part 3 .	151
Figure 67.	Test 6: Scenario 2 Packet Capture on NAT 1 (pinged from NAT 2), Part 4 .	152
Figure 68.	Test 6: Scenario 2 Packet Capture on Router (pinged from Router)	154
Figure 69.	Test 6: Scenario 2 Packet Capture on NAT 1 (pinged from Router).....	155
Figure 70.	Test 6: Scenario 3 Packet Capture on Router (pinged from NAT 2).....	160
Figure 71.	Test 6: Scenario 3 Packet Capture on NAT 1 (pinged from NAT 2), Part 1 .	161
Figure 72.	Test 6: Scenario 3 Packet Capture on NAT 1 (pinged from NAT 2), Part 2 .	162
Figure 73.	Test 6: Scenario 3 Packet Capture on NAT 1 (pinged from NAT 2), Part 3 .	163
Figure 74.	Test 6: Scenario 3 Packet Capture on Router (pinged from Router)	165
Figure 75.	Test 6: Scenario 3 Packet Capture on NAT 1 (pinged from Router).....	166
Figure 76.	Test 6: Scenario 4 Packet Capture on Router (pinged from NAT 2), Part 1 .	171
Figure 77.	Test 6: Scenario 4 Packet Capture on Router (pinged from NAT 2), Part 2 .	172
Figure 78.	Test 6: Scenario 4 Packet Capture on NAT 1 (pinged from NAT 2)	173
Figure 79.	Test 6: Scenario 4 Packet Capture on Router (pinged from Router), Part 1 .	175
Figure 80.	Test 6: Scenario 4 Packet Capture on Router (pinged from Router), Part 2 .	176
Figure 81.	Test 6: Scenario 4 Packet Capture on NAT 1 (pinged from Router).....	177

LIST OF TABLES

Table 1.	Hardphones	7
Table 2.	VoIP Protocols	9
Table 3.	SIP and H.323 components [From Ref. 9].....	19
Table 4.	Basic Call Control Features	23
Table 5.	H.235 Security Profiles	28
Table 6.	SIP Security Features.....	29
Table 7.	Summary of Hardphones	45
Table 8.	Summary of Softphones.....	46
Table 9.	Test 6: Test Scenario Configurations.....	129
Table 10.	Test 6: Ping Operations.....	130
Table 11.	Test 6: Scenario 1 Result	136
Table 12.	Test 6: Scenario 2 Result	147
Table 13.	Test 6: Scenario 3 Result	159
Table 14.	Test 6: Scenario 4 Result	170

THIS PAGE INTENTIONALLY LEFT BLANK

ACRONYMS AND ABBREVIATIONS

AP	Access Point
ASN.1	Abstract Syntax Notation One
ATA	Analog Telephone Adapter
BES	Back End Service
CODEC	Coder-Decoder
DNAT	Destination Network Address Translation
DoD	Department of Defense
DoS	Denial of Service
GSM	Global System for Mobile Communication
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPsec	Internet Protocol Security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ITU-T	International Telecommunications Union – Telecommunications Standards Committee
LAN	Local Area Network
MEGACO	Media Gateway Control
MIME	Multipurpose Internet Mail Extensions
MGCP	Media Gateway Control Protocol
MCU	Multipoint Control Unit
MYSEA	Monterey Security Architecture
NAT	Network Address Translation
NIST	National Institute of Standards and Technology
PBX	Public Branch Exchange
PDA	Personal Digital Assistant
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RTCP	Real-time Control Protocol
RTP	Real-time Transport Protocol

SDP	Session Description Protocol
SIP	Session Initiation Protocol
S/MIME	Secure/ Multipurpose Internet Mail Extensions
SNAT	Source Network Address Translation
SS7	Signaling System 7
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol
WAN	Wide Area Network
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network

ACKNOWLEDGMENTS

I would like to thank my thesis advisors, Cynthia Irvine and Thuy Nguyen, for their support and guidance throughout the thesis process. I would also like to thank Jean Khosalim and Phil Hopfner for lending me a helpful hand during testing.

This material is based upon work supported by the National Science Foundation under Grant No. DUE-0114018 and the Office of Naval Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation or the Office of Naval Research.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION

Voice over Internet Protocol (VoIP) is becoming a popular technology. Currently, three million households use VoIP. It is estimated that the number will increase to twenty-seven million by the end of 2009 [1]. Furthermore, companies and public-sector organizations are expected to invest over nine hundred million dollars into the VoIP technology in 2005, an increase of more than two hundred million dollars compared to last year [2]. This indicates that VoIP continues to grow in popularity and remains a promising telecommunication technology that will gradually replace traditional PSTN phone systems.

Integration of VoIP capabilities into the existing MYSEA architecture is highly desirable for both economical and management reasons. Deploying VoIP greatly reduces the cost of making long distance calls from the MLS LAN to an external network such as the Internet. Management of telephone systems in the MLS LAN will also be simplified with the use of VoIP as rewiring phones for nomadic users is no longer needed. Thus, extending MYSEA to include VoIP is beneficial.

B. PURPOSE OF STUDY

This study is a preliminary step in integrating VoIP capabilities into the existing MYSEA architecture. The objective is to determine the feasibility of this integration. The current MYSEA environment has a number of NAT components that may make the integration of VoIP into the MLS environment difficult if not impossible. A set of experiments were devised and conducted to test if VoIP works with the NAT components in MYSEA.

C. ORGANIZATION OF PAPER

This paper is organized into five chapters and six appendices. A brief introduction is provided in Chapter I. Chapter II provides the background information related to this research study. A technical comparison between two popular VoIP protocols, H.323 and SIP, is presented in Chapter III. One of the two protocols will be selected for testing purposes. Chapter IV describes the test plan used to confirm the

feasibility of integrating VoIP capabilities into MYSEA. The problems encountered and results from each test are also discussed in this chapter. The last chapter or Chapter V talks about future work and conclusion.

Six appendices are also included in this paper. Appendix A has surveys of hard and softphones. Appendices B through F contain descriptions, instructions, and results of tests described in Chapter IV.

II. BACKGROUND

This chapter presents background information pertaining to this thesis study. It includes a high-level technical overview of the current VoIP technology. Finally, the MYSEA architecture is described in the last section.

A. INTRODUCTION

The traditional phone system has been evolving since the first voice transmission in 1876 using a ring-down circuit. Today, phone systems no longer run on an analog network, instead they use a digital network known as the Public Switched Telephone Network (PSTN). The PSTN greatly reduces the amount of noise inflicted by analog voice amplification during transmission and provides number of services, such as call waiting, call forwarding, three-way calling, call blocking, etc. Despite the benefits obtained from the PSTN, there are drawbacks to the system that motivated the VoIP solution.

1. Advantages of VoIP

VoIP has many advantages over the traditional PSTN phone systems. These include the efficient use of bandwidth, reduction or possible elimination of long distance and phone charges, convergence of the voice and data networks, and advanced features. Some of them will be discussed further below.

a. Efficient Use of Bandwidth

Bandwidth is a key performance measure of a network. It defines how many bits can be transmitted every second, which means the more bandwidth available, the more data can be sent in a given period of time.

PSTN phone system requires a minimum of 64-kbps of dedicated circuit between the two calling devices. The circuit is reserved for the entire duration of the call regardless of whether or not any data is in transmission. Hence, bandwidth is unnecessarily wasted. On the other hand, VoIP uses IP networks that have the flexibility to allocate bandwidth as needed and reserve the unallocated bandwidth for other data. Thus, the use of network bandwidth in VoIP is more efficient.

b. Reduction or Possibly Elimination of Long Distance and Phone Charges

The cost of a long distance call generally depends on two factors: duration and destination of call. Charges can accumulate when an enterprise or individual frequently makes this type of call. VoIP service providers have monthly flat-rate plans that offer unlimited or fixed-number of minutes to make calls, including long distance calls. These plans are much more economical than the traditional charge-by-minute service. Thus will greatly reduce or possibly eliminate the phone and long distance charges for individuals and enterprises that make frequent long distance phone calls.

c. Convergence of Voice and Data Networks

Traditionally, a voice network only transmits voice and a data network only carries data. This is no longer true. Data makes up major traffic on voice networks. Unlike data networks, voice networks are not efficient in carrying data due to its inflexible bandwidth allocation and limited bandwidth. Therefore, most enterprises maintain both networks.

However, in many cases, management and maintenance of two different networks has proved to be cumbersome and costly for enterprises. Upgrading the voice network equipment such as the Public Branch Exchanges (PBX) telephones burdens enterprise budgets. If VoIP is deployed, the voice network will no longer be needed and will leave the enterprise with only the data network.

d. Advanced Features

VoIP provides all the services of a traditional phone system including speed-dial, call waiting, busy signaling, caller-ID, etc. In addition, VoIP can interoperate with services traditional phone systems lack such as video-conferencing, instant messaging, email, click-to-dial, and directory service.

2. Disadvantages of VoIP

While more home users and businesses are in the transition to use VoIP, the technology does have shortcomings. These will be discussed in detail below.

a. Quality of Voice

Voice data traveling across an IP network is highly susceptible to delay and loss due to routing and network latency. Voice in analog form has to be converted

and compressed into digital packets before transmission over an IP network. A compression method that aggressively minimizes the size of voice packets will deteriorate the quality of voice. As a result, the quality of voice using VoIP may be worse than that obtained from PSTN due to delay, loss, and compression of the information.

b. Security

In many cases, maintaining separate voice and data networks can be difficult and costly. Convergence of both networks simplifies management and greatly reduces cost. However, convergence leads to security problems. Voice will be vulnerable to the same attacks as other data traveling across an IP network. Attacks include interception, modification, spoofing, man-in-the-middle attacks and denial of service.

c. Availability

Making a VoIP call requires a connection to an IP network through properly configured network devices that are dependent on a stable electrical power supply. Power outage and connection problems will prevent an individual from making or receiving a VoIP call.

d. 911

Currently, none of the VoIP protocols provide information regarding the caller's physical location to the emergency operator. When the caller dials 911, there is no guarantee the call will be routed to the nearest 911 police station. At the same time, the 911 operator has no way of identifying the location of the caller.

B. VOIP OVERVIEW

VoIP uses IP or a packet-switched network as the data transmission vehicle. A VoIP system digitizes voice using an audio codec, divides the digitized voice into packets, and sends the packets over an IP network to the destination. All packets are routed without a guarantee that they will travel the same path. Unlike a PSTN call, no dedicated circuit is ever created for a VoIP call.

The exact process required to set up a VoIP call is dependent on the VoIP protocol. Two types of protocols are necessary to complete a VoIP call: signaling and media transport. A signaling protocol has the responsibility of establishing a session

between the call participants. A media transport protocol specifies the rules and formats of the actual voice packets. Currently, the Real Time Protocol is commonly used as the media transport protocol in VoIP. However, there is a wider variety of signaling protocols. VoIP protocols will be further discussed later in this chapter.

For PSTN systems, a phone number consisting of digits is used to locate a phone. A phone number in VoIP can be a regular PSTN phone number, an address, or an alias. The “phone number” ultimately is translated to a 32-bit or 128-bit IP address depending on whether IPv4 or IPv6 is used. Every VoIP signaling protocol must provide address resolution capability.

There are four general VoIP communication modes. They are Phone-to-Phone, Phone-to-PC, PC-to-Phone, and PC-to-PC.¹ Voice transmission is carried by both PSTN and IP networks under the first three modes. A VoIP service provider that interconnects the PSTN and VoIP networks is needed for the first three modes when a call originates from a PSTN network and arrives at a VoIP network or vice versa. Voice travels exclusively across the IP network in the fourth mode.

1. VoIP Phone Overview

VoIP phones generally fall into three categories: PSTN phones, hardphones, and softphones. A PSTN phone is the type of phone almost every household has and is connected to a phone jack using a telephone cable. Technically, a PSTN phone in itself is not a VoIP device, but it can be used to make VoIP calls with the use of a phone adapter known as the Analog Telephone Adapter (ATA) that converts voice from analog to digital form.

A hardphone, also commonly known as an IP phone, can look identical to a PSTN phone. It is an independent device that understands VoIP protocols. Unlike a PSTN phone, a hardphone does not require an external device such as an ATA to make VoIP calls. All it needs is an Internet connection. Table 1 lists various types of hardphones and their corresponding descriptions from [3].

¹ Phone refers to a traditional phone and PC refers to a personal computer.

Hardphone Type	Description/Characteristics
Ethernet	Has an Ethernet port Connects directly to the IP network
Cordless	Has IP interface on base stations
WLAN or WiFi	Has built-in WiFi transceivers Connects to a WiFi base station
WLAN/WiFi and GSM	Same as WLAN/WiFi phones but can also transfer calls to GSM network
Voice and Video	Supports for both voice and video

Table 1. Hardphones

A softphone is a VoIP phone in the form of software. A softphone runs on a computing device such as a desktop, laptop, or PDA, and is typically Operating System dependent. This type of phone needs audio support such as speakers and microphones for communication purposes. Appendix A has a survey of hard and softphones.

C. VOIP SERVICES

The VoIP technology is made up of four distinct services: signaling, encoding, transport, and gateway control [4]. A signaling VoIP protocol establishes and manages a connection between the endpoints when a call is made. Signaling protocols are discussed in Section D. When the conversation takes place, voice has to be encoded before it is transmitted over the IP network. The encoded voice packets will then be transported via the IP network to the destination. A gateway may be needed to convert voice into another format suitable for the receiving network. For example, a gateway will convert voice from digital PSTN to digital IP form when voice packets come into an IP network from a PSTN network.

1. Encoding

Voice, in its native form, cannot be transmitted over an IP network. A voice codec is used to convert voice from analog to digital data or digital to analog data, compress voice to optimize bandwidth usage, and packetize the voice data in preparation for transmission. A codec determines bandwidth usage and quality of voice. Higher quality voice transmission usually requires more bandwidth. The tradeoff between the two factors is critical when deciding what codec to use in VoIP applications. Various codecs exist to support VoIP. However, the three most commonly used codecs are G.711, G.723.1, and G.729A. More information on the above codec specifications can be found on the ITU-T website.

2. Transport

Media transport protocols such as Real Time Protocol (RTP) deliver the encoded voice packets over an IP network. RTP is a standard developed by Internet Engineering Task Force (IETF) to transport real-time audio and video data. RTP does not guarantee reliable transmission of packets. It usually runs on top of UDP due to the delay-intolerance of voice conversation and uses a dynamically assigned UDP port in the range 1024 – 65535.

The RTP Control Protocol (RTCP) is the control counterpart of RTP. RTCP, also developed by IETF, is not required to be used with RTP. However, RTCP can be used to monitor transmission performance. End users in a VoIP session can send transmission statistics in RTCP packets upon receiving packets. This information is useful to determine network and delivery performances and keep track of retransmission needs. Similar to RTP, RTCP uses a dynamically assigned UDP port.

3. Gateway Control

A gateway connects the PSTN and VoIP networks. Voice packets arriving at its IP interface will be converted by the gateway from a format understandable by IP to one that is understandable by PSTN and vice versa. A gateway is necessary for communications between Phone-to-Phone, Phone-to-PC and PC-to-Phone.

D. VOIP PROTOCOLS

Table 2 lists some well-known VoIP signaling protocols. A VoIP signaling protocol defines the formats of VoIP messages and rules for message exchange necessary

to establish a VoIP call. The signaling protocol is responsible for setting up a VoIP call, which includes tasks like locating users and negotiating session parameters between the two end devices. A media gateway control protocol control communication amongst the gateways in an IP networks.

Protocol	Organization	Type
H.323	ITU-T	Signaling
Session Initiation Protocol (SIP)	IETF	Signaling
MGCP	ITU-T	Signaling
Megaco/H.248	ITU-T/IETF	Signaling

Table 2. VoIP Protocols

1. H.323

H.323 is an open standard developed by ITU-T in 1996. H.323 was originally designed for multimedia conferencing and was later extended to support VoIP. H.323 is a suite of protocols that provide services such as end-to-end multipoint conferencing, audio and video codecs, management and accounting, and security. Since 1996, the protocol has undergone a series of changes, with the latest version (H.323v4) providing many enhanced feature and services. Refer to Chapter III for more details.

2. Session Initiation Protocol

The Session Initiation Protocol is developed by IETF. It is an application protocol designed to establish a two-way communication session. SIP is gaining popularity in the VoIP market despite the fact that it is a fairly young protocol developed in 1998. SIP is generally more scalable, simple, and extensible than H.323. Some believe that SIP will eventually become the official VoIP signaling protocol standard. Refer to Chapter III for more details.

3. MGCP

Media Gateway Control Protocol (MGCP), developed by IETF, controls communication among VoIP gateways in an IP network. Two components exist in the MGCP architecture: call agents, also known as media gateway controller, and gateways.

MGCP is a master-slave protocol in which the master call agent sends signaling, control, and processing commands to the gateway. The gateway acts as a slave and executes the commands sent by the call agent. MGCP does not replace SIP or H.323. Rather, the protocol is used to manage signaling and control activities for VoIP network gateways such as H.323, SIP, and SS7 signaling.

4. Megaco/H.248

Megaco/H.248, developed jointly by ITU-T and IETF, has the same architecture as MGCP. However, Megaco/H.248 offers several advantages over MGCP such as the support of multimedia and multipoint conferencing enhanced services, improved syntax for more efficient semantic message processing, TCP and UDP transport options, support for both text and binary encoding, and formalized extension process for enhanced functionality [5].

E. VOIP CHALLENGES

1. Quality of Service

Quality of Service (QoS) is not a major concern in PSTN systems because a fixed amount of bandwidth is dedicated to a call and transmission of voice follows the same circuit for the duration of the call. On the other hand, when a VoIP call is made, digitized voice will be transmitted using an IP network that has no fixed-bandwidth allocation mechanism. Thus, it is subject to jitter, latency, and packet loss problems, of which VoIP is intolerant.

There are many good reasons to deploy VoIP, however, it makes no sense to use VoIP if the quality of a VoIP call is lower than a traditional PSTN phone call. QoS must be addressed to an acceptable level such that end users can carry on a smooth conversation with minimal interruptions

2. Security

Security is another important aspect of VoIP. Convergence of voice and data networks means that both voice and data have to be protected. Eavesdropping a pure PSTN phone conversation is more difficult than a VoIP conversation, because interception of a regular PSTN call requires physical access to the phone lines or compromise of the corporate PBX. On the other hand, a VoIP conversation can be intercepted by an adversary anywhere along the path where the digitized voice packets

travel. The security problem is intensified when sensitive personal information such as social security and credit card numbers are given out over a VoIP call.

The transmission of voice over an IP network is subject to security risks. It is important to ensure the confidentiality, integrity, and authenticity of a VoIP conversation and availability of resources when a VoIP call needs to be placed. In summary, the conversation and the VoIP network resources are the two main assets that require protection. Security mechanisms must be used to prevent both internal and external eavesdropping, spoofing, replay, and denial of service attacks. Many security mechanisms such as encryption, firewalls, and Network Address Translation (NAT) exist to address these threats. However, almost every one of them raises problems or affects the overall performance of a VoIP in some ways.

Encryption effectively protects the confidentiality and possibly authenticity of VoIP packets by making it impossible for people other than the intended recipient to read the packets. However, encrypting every packet, at the sending end and decrypting it at intermediate nodes and at the receiving end could cause an immense amount of delay, thus lowering the QoS. The size of an encrypted packet is often bigger than the plaintext packet, thus requiring more bandwidth and leading to a possibility of packet drop. This is a typical tradeoff between security and performance.

A firewall is often the first layer of defense in securing a network. It sits between the internal and external network, inspects every incoming and outgoing packet, and blocks those packets that it thinks is malicious. Firewalls usually inspect packets by examining certain fields, such as IP addresses, ports, and protocol type, in the packet headers. However, some VoIP protocols such as H.323 use dynamic ports to send or receive messages. A stateless firewall that only looks at header information to determine packet admissibility might drop some of the messages. To ensure the admission of those messages, the stateless firewall would have to open many ports and leave itself in a vulnerable statue. A stateful firewall, one that stores information about a session along with previous packet transactions, can inspect a packet's application layer data and can manage the dynamic port problem. However, a stateful firewall introduces latency due to the extensive packet inspection. As a result, network performance may not be optimal.

Network Address Translation (NAT) is a method of mapping a group of private IP addresses to a group of public network IP addresses. NAT conserves IP addresses by sharing a limited number of public IP addresses among many internal hosts. A public IP address can be mapped to multiple internal hosts. Furthermore, NAT hides internal IP addresses from the outside world so that adversaries outside the network cannot directly attack internal hosts. VoIP signaling and media transport protocols often use different ports. Furthermore, RTP and RTCP use random ports to exchange data, thus complicating the NAT process. When NAT receives the actual digitized voice packets, it has no knowledge of where to send it. As a result, the packets may get dropped.

Similar to the use of encryption, the use of firewalls and NAT will also affect QoS because every packet coming in will have to be processed to determine admission. The uses of encryption, firewall, and NAT are just three examples of defense mechanisms against possible VoIP threats. However, these defenses often cause problems in the operation of VoIP processing and performance.

F. NAT

Network Address Translation (NAT) is primarily used for two purposes: public IP address conservation and security. The advantage of using NAT is that any number of internal hosts using un-routable private IP addresses can be connected to another network, such as the Internet, using a small number of public IP addresses. When an internal host wants to communicate with an external host, NAT maps the private IP address of the internal host to one of its un-used public IP addresses. The process of NAT rewriting the private IP address with a public IP address in the source IP address field of all packets initiated from a local host is known as Source Network Address Translation (SNAT). Destination Network Address Translation (DNAT) refers to the process of NAT rewriting the public IP address with a private IP address in the destination IP address field of all packets received at its public interface. Note that a NAT device must have routing capabilities to route packets in and out of two different networks. SNAT and DNAT allow an internal host to communicate with an external host without ever exposing its internal IP address. This mechanism of hiding internal IP address provides another layer of protection for system security.

1. netfilter/iptables

Any Linux system can be turned into a NAT device using *netfilter* and *iptables*. These two open source kernel modules are included in most Linux distributions that provide networking functions including NAT, routing, and firewall. Furthermore, *iptables* has a powerful connection tracking mechanism that allows it to associate packets with their corresponding sessions. This mechanism is essential in stateful firewalls. More information about *netfilter* and *iptables* can be found at [6]. The next section describes a component in the MYSEA architecture that uses *netfilter* to do NAT.

netfilter/iptables consults the *nat* table to determine if the IP address and/or port of a packet needs to be rewritten and how those fields should be rewritten. The *nat* table contains three chains of rules. Two of them are PREROUTING and POSTROUTING. The PREROUTING chain is referenced when a DNAT decision has to be made while the POSTROUTING chain is used to make SNAT decisions. The following are two sample NAT rules:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 192.168.0.1
```

```
iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 192.168.1.10
```

The first rule instructs the NAT device to modify the source IP address of all outgoing packets to 192.168.0.1 after the packet is processed by the routing logic. The second rule tells the NAT device to rewrite the destination IP address of all incoming packets to 192.168.1.10 before routing logic takes place.

2. SIP with NAT

SIP-based VoIP calls rely on two protocols: SDP for negotiating of session parameters and RTP for transporting of voice data. Two endpoints setup a VoIP connection with exchange of the INVITE and 200 OK messages. Each message has SDP information embedded in the payload specifying the IP address the endpoint expects to receive RTP voice packets. Before sending out either the INVITE or 200 OK packet during the call setup, an endpoint located behind a NAT device writes its private IP address as part of the SDP data. The NAT device, a layer three device, performs SNAT to the message it receives from the endpoint and then sends the packet to the destination address. Since SDP is a layer five protocol, the NAT device is unable to examine and

rewrite the private IP address embedded in the SDP section of the message. When the other endpoint wants to send RTP packets, it will send it to the private IP address indicated in the SDP portion of the received message and hence the RTP packets will get dropped.

G. MYSEA OVERVIEW

The Monterey Security Architecture (MYSEA) is a multi-level distributed operating environment designed to allow secure access to information at different classifications. MYSEA consists of a combination of commercial-off-the-shelf (COTS) and high assurance components. The COTS components are used to perform common user tasks whereas the high assurance applications are used to enforce security policies. Such a design is especially advantageous to organizations such as the DoD that invest heavily on COTS products but has a need to manage information with different sensitivities [7].

Figure 1 is an illustration of the MYSEA network architecture. Communication between an untrusted client on the MLS LAN and another client is mediated by a MLS server running XTS-400. The MLS server enforces security policies and provides a number of security-related services. Each MLS LAN client communicates with the MYSEA server via an inline Trusted Path Extension (TPE). The TPE establishes an encrypted trusted path and negotiates session level information with the MLS server on behalf of the client. Each TPE is also a NAT device that hides the internal IP address of the clients. Currently every TPE on the MYSEA testbed has a unique private IP address whereas every client uses the same private IP addresses. Refer to [8] for more detail discussion of MYSEA.

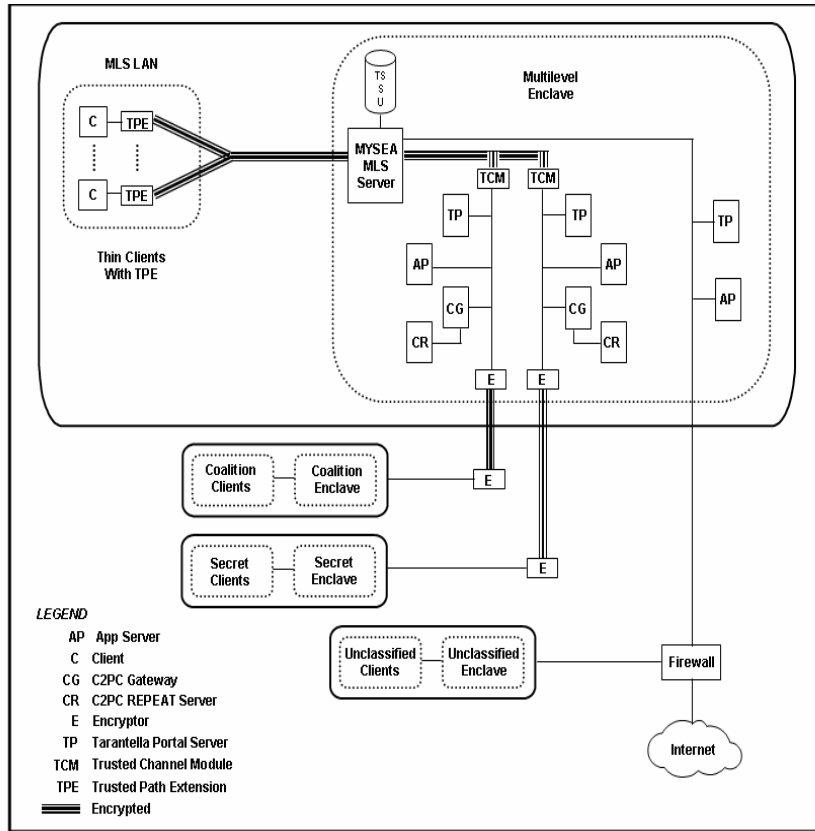


Figure 1. MYSEA Network Architecture [From Ref. 8]

H. SUMMARY

This chapter presents background information that is relevant to this project. To prepare for testing, a VoIP protocol must be selected as the test protocol. The next chapter compares two popular VoIP protocols, namely H.323 and SIP, and ends with a protocol selection for testing.

THIS PAGE INTENTIONALLY LEFT BLANK

III. TECHNICAL COMPARISON BETWEEN H.323 AND SIP

The purpose of this chapter is to compare two dominant VoIP signaling protocols: H.323 and Session Initiation Protocol (SIP). At the end of the study, one of the two protocols will be selected for testing. Selection is based on the protocols' simplicity and flexibility, extensibility, scalability, and security. This chapter consists of two sections. The first section describes the protocols whereas the second section is a summary of two SIP and H.323 comparisons presented in [9] and [10].

A. H.323 AND SIP OVERVIEW

H.323 and SIP are the front-runners in the VoIP industry. H.323 came about in 1996, two years before the birth of SIP in July 1998. H.323 is still widely used in enterprises and continues to be improved, while SIP is gaining popularity and undergoes more development. The following subsections present a high level overview of the two protocols.

1. Background

H.323v1 was developed by ITU-U as a “standard for real-time video-conferencing over non-guaranteed quality of service LANs” [9]. The standard has undergone several revisions. The latest version, H.323v4, defines basic call control and signaling for multimedia applications. The protocol is specifically designed to support multimedia and voice applications. It was extended to support VoIP. The ITU-U initially concentrated on developing multimedia functionalities, supplementary services, and internetworking capabilities into H.323. As those capabilities become standardized, ITU-U works to address the protocol's security, QoS, and mobility issues.

SIP, IETF's standard for establishing VoIP connections, was standardized in 1999 and revised in 2002. SIP is an application layer protocol designed to setup, modify, and tear down generic sessions. Other fundamental services it provides include user location, session invitation and session negotiation. As with all IETF protocols, SIP was not developed to support a particular type of application. Rather, SIP is designed to work with any application that may need its services. The IETF initial focus was on standardizing the protocol to support session initiation. Currently, a large amount of

effort is placed on defining specific applications, such as internetworking with legacy networks and providing supplementary services [9].

2. Architecture

Both H.323 and SIP have both peer-to-peer and client-server architectures. H.323 specifies a complete framework that defines the protocols and the message flows for multimedia communications. The standard covers all phases of a VoIP call including set-up, call control, and media transport. Other issues critical to the quality of a VoIP call such as QoS, security, and mobility are also addressed in the standard.

H.323 is actually a suite of protocols that can be broken down into six classes: call control and signaling, audio processing, video processing, data conferencing, media transportation, security, and supplementary services. Information regarding the different protocols used in H.323 can be found in [11].

Figure 2 depicts the H.323 protocol stack. The lighter colored blocks represent optional components whereas the darker colored blocks represent mandatory components necessary to complete a VoIP call. It is important to note that both H.323 and SIP rely on the support of several common protocols such as TCP/UDP, IP, RTP and RTCP and audio processing services. In summary, only H.245, H.225.0/Q.931, and H.225.0/RAS are essential to achieve the signaling part H.323 VoIP call.

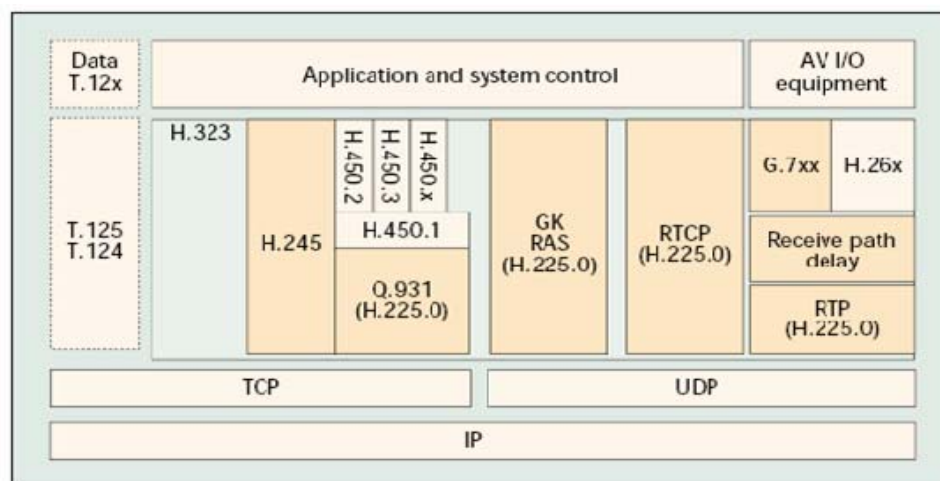


Figure 2. H.323 Protocol Stack [From Ref. 9]

SIP by itself only defines setup and teardown of sessions. Advance signaling features are specified as SIP extensions. Furthermore, QoS and mobility are not addressed by SIP but can be supported by other protocols. SIP depends on the Session Description Protocol (SDP) to describe parameters for multimedia session between two endpoints. SDP is a text-based media-description format that is carried in SIP messages. Figure 3 depicts the SIP protocol stack. Again, the darker component is essential to the signaling part a SIP VoIP call.

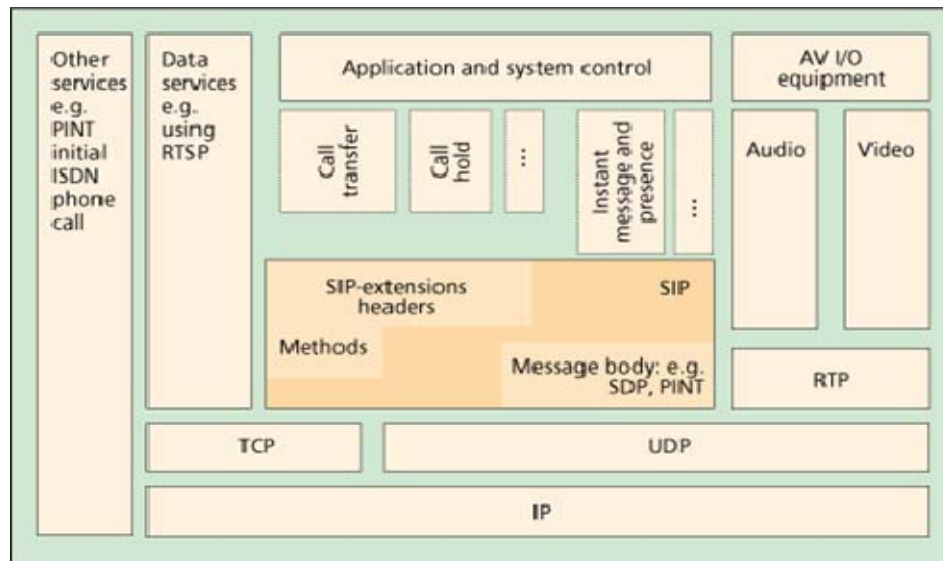


Figure 3. SIP Protocol Stack [From Ref. 9]

3. Components

H.323 and SIP divide functions to components in a similar fashion. Basic call function controls are assigned to the terminals whereas services requiring network support are assigned to network servers. Table 3 lists SIP and H.323 network components by types.

	Client	Servers in the network		
SIP (IETF)	Terminal	Proxy server, registrar	Conference server ¹	Gateway ^{1,2}
H.323 (ITU-T)	Terminal	Gatekeeper	MCU	Gateway
¹ Not standardized up to now. ² Addressed in several drafts, e.g., SIP-T MCU: Multipoint control unit. Gateway: maintains transition to traditional telephony.				

Table 3. SIP and H.323 components [From Ref. 9]

An H.323 network consists of terminals and a gateway. A gatekeeper, Multipoint Control Unit (MCU), and Back End Service (BES) may also be deployed as part of the network. A terminal is any VoIP-enabled device. The gateway provides translation services for terminals that use different communication protocols including non-VoIP protocols such as PSTN. A gatekeeper performs address translation, controls accesses of terminals, manages bandwidth and makes routing decisions. Although a gatekeeper is optional, it is often an important component in an H.323 network because of the services it provides. A H.323 network can have a MCU that facilitates communication among multiple endpoints. A Back End Service usually exists to support the gatekeeper by maintaining information about endpoints such as the endpoints' permissions, configurations, and services [5].

A SIP network has endpoints, a proxy server or redirect server, location server, and a registrar. The registrar authenticates users and stores location information from users. A proxy server, which can be integrated with the registrar, resolves addresses and forwards messages on behalf of the endpoint to another proxy server or the destination endpoint during call setup and teardown. The redirect server performs tasks similar to those of a proxy sever but instead of forwarding the message, the redirect server sends the resolved address back to the endpoint and lets the endpoint communicate directly with the other endpoint. The location server supports the registrar by maintaining location information of endpoints [5].

4. Call Setup

Call setup refers to the actions necessary to establish a connection between two endpoints. This process must be completed before the endpoints can exchange actual voice data. The following subsections describe simple H.323 and SIP call setups. The scenarios assume that Alice initiates a non-local call to Bob.

a. H.323

H.225.0/RAS, Q.931 in H.225.0, and H.245 are necessary to establish an H.323 VoIP connection. These protocols provide functions necessary for call registration, call setup, and capability exchange. Figure 4 illustrates a simple H.323 call setup.

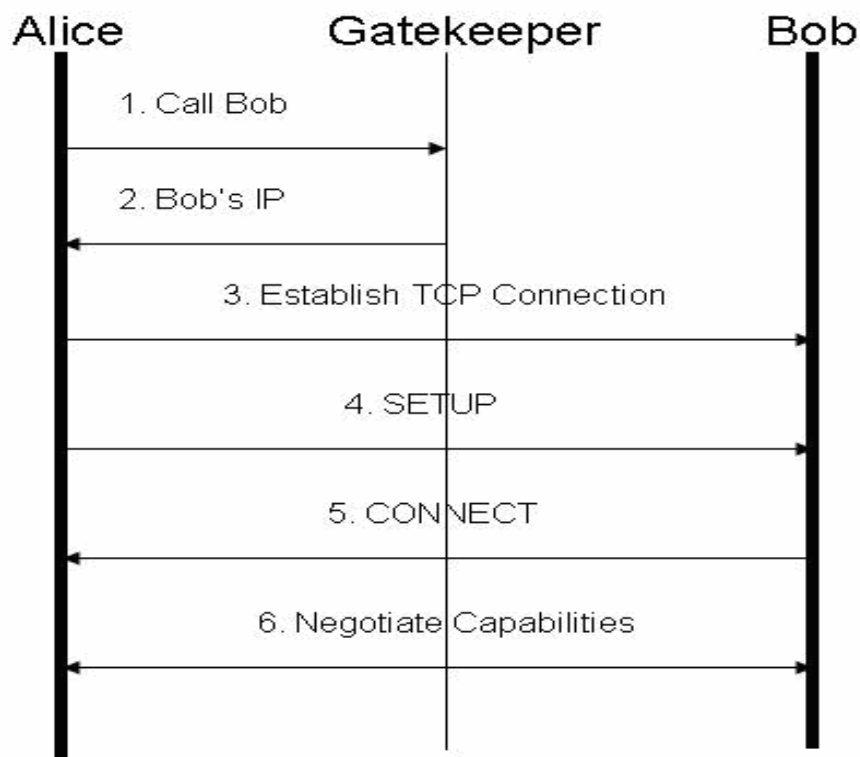


Figure 4. Simple H.323 Call Setup

When Alice dials Bob's phone number, (Step 1) Alice's terminal sends a Registration Admission Request to the gatekeeper using H.225.0/RAS. The gatekeeper registers Alice into the system, admits and grants resources to Alice and finds Bob's IP address. Next, (Step 2) the gatekeeper sends the IP address to Alice. Alice then establishes a TCP connection with Bob at the IP address she received (Step 3). Alice sends a SETUP message to Bob using Q.931/H.255.0, an ISDN-connection control protocol (Step 4). Bob sends Alice back a CONNECT (Step 5) message to Alice using the same protocol indicating acceptance to the connection. Finally, Alice and Bob negotiate terminal capabilities using H.245 (Step 6). Then H.245 will open logical channels for both endpoints to start the conversation.

b. SIP

Figure 5 illustrates a simple SIP call setup. In this example, an integration of the registrar into the proxy server is assumed. Before Alice calls Bob, Alice's terminal must register itself with the registrar (Step 1). This step is similar to the first step in the

H.323 simple call setup. After the registration is completed, Alice may call Bob by sending the proxy server an INVITE Bob message (Step 2). The proxy server looks up Bob's IP address and forwards the invitation to Bob (Step 3). An OK response will be received by the proxy server from Bob indicating acceptance to the call (Step 4) and the response will in turn be forwarded to Alice (Step 5). Throughout this process, session parameters and terminal capabilities are transparently exchanged inside the INVITE and OK messages from both parties using SDP or some other methods. From now on, the two parties may communicate in a peer-to-peer fashion.

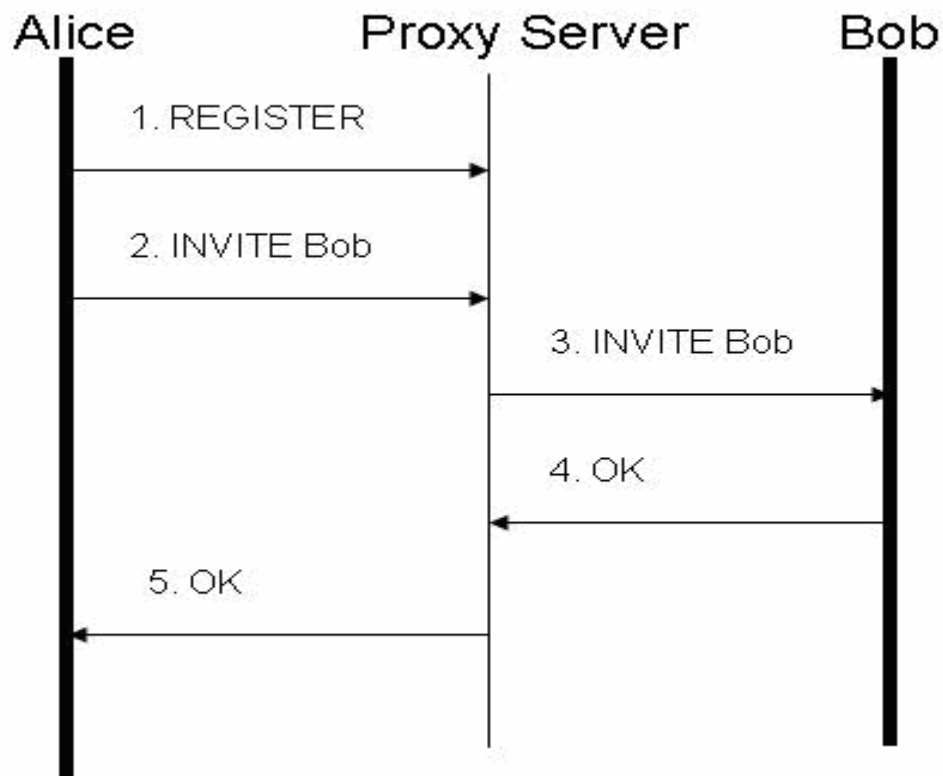


Figure 5. Simple SIP Call Setup

5. Services

H.323 and SIP both provide basic call controls as well as advanced features. Table 4 lists the features common to both protocols.

Feature	H.323	SIP
Call Setup	Yes	Yes
Call Teardown	Yes	Yes
Call Waiting	Yes	Yes
Call Hold	Yes	Yes
Call Transfer	Yes	Yes
Call Forwarding	Yes	Yes
Call Return	Yes	Yes
Call Identification	Yes	Yes
Call Park	Yes	Yes
Capabilities Exchange	Yes	Yes

Table 4. Basic Call Control Features

B. H.323 AND SIP COMPARISON

H.323 and SIP are different in many ways despite the fact that they provide similar call control services and are widely used in VoIP applications. One of their fundamental differences lies in their original intents and designs. H.323 was designed with a focus on multimedia and voice communications whereas the SIP design focused on providing only session initiation services. H.323 uses a top-down approach to specify a complete framework for providing multimedia and voice services. It is telecommunication-oriented as it uses existing multimedia protocols in the ITU H-series to support provide various services. SIP, on the other hand, uses a bottom-up approach. Its modular design allows it to work with a wide range of applications. SIP takes on an Internet-oriented design by adopting a number of features from HTTP and SMTP, two of the most successful Internet protocols [9, 10].

Their implementation approaches lead to differences in the simplicity and flexibility, extensibility, scalability, and security of the protocols. The following subsections examine these differences.

1. Simplicity and Flexibility

Simplicity and flexibility are two important measurements to determine the quality of the protocols' design. This subsection compares H.323 and SIP based on these two aspects. Protocol specification, message encoding, and protocol interactions will be closely examined.

a. Protocol Specification

SIP is simpler in nature than H.323. According to [9], a SIP-based VoIP implementation can be done with four headers (To, From, Call-ID, and Cseq) and three types of requests (INVITE, ACK, and BYE).

H.323, on the other hand, consists of numerous protocols such as H.225.0 for call signaling, H.245 for call control, H.323 for conferences, H.450.1 to H.450.9 for supplementary services, H.235 for security and encryption, etc. Many services require a number of H.323 protocols to interact with each other. This further intensifies H.323's complexity problem [10].

b. Message Encoding

H.323 messages are encoded by the ASN.1, an international standard used to specify data used in communication protocols. Since ASN.1 messages exist in binary form, a special tool is needed to parse the messages. SIP adopts the HTTP tradition by using text-based messages. Hence, SIP messages are generally easy to parse, generate, and debug [10].

c. Protocol Interactions

H.323 is complicated because of the many of protocols it encompasses. A number of protocols are often required for a single service in H.323. For example, connection establishment, as illustrated in Figure 3, requires Q.931/H.255.0, H.225.0/RAS, and H.245. On the other hand, SIP uses a single INVITE request to establish the connection even though it depends on SDP to negotiate session parameters. H.323 also allows those three protocols to be used in different orders to establish a

connection. Thus, network devices such as firewalls, endpoints, gatekeepers, and gateways must support and understand all three connection establishment methods [12].

d. Applications

SIP is designed to create, manage, and tear down generic sessions. As a result, SIP has the flexibility to work with a wide range of applications. Voice and multimedia are just two applications of SIP. Others include voice-enriched e-commerce, web page click-to-dial, Instant Message, and IP Centrex services. H.323 was designed to focus on a specific type of communication, namely voice and multimedia conferencing. Thus, its applications are not as wide as SIP's. ITU-U is currently working toward providing non-VoIP services in H.323 [9].

2. Extensibility

Extensibility defines how easy it is to add new features to the existing protocols. This aspect of the protocols will be evaluated based on extensible mechanisms, backward-compatibility, and interoperability.

a. Extensible Mechanisms

Both H.323 and SIP have certain extensible mechanisms. H.323 has nonstandardParm fields in its ASN.1 messages. Each nonstandardParm field is identified by a unique vendor code and the information contained in this field is only meaningful to the specific vendor. These fields allow vendors to add extensions [10].

SIP adopts the HTTP's use of hierarchical numeric warning codes. Its warning codes are represented by three digit numbers where the first digits identify which of the six categories the codes belong to.² The list of warning codes can be easily extended under this hierarchical structure system. SIP features are also extensible. New SIP features can be officially added by registering the names of the features with the Internet Assigned Numbers Authority (IANA). New features will not cause confusion on the server side because SIP specifies that a server shall ignore the request of a feature in a SIP header if the server does not understand or support the feature.

² SIP warning codes are divided into six categories: provisional, redirection, request and server failure, busy everywhere responses, successful and bad requests.

b. Backward-Compatibility

H.323 supports full backward compatibility among different implementation versions. In other words, obsolete features have to be carried over from one version to the next. Hence, the code can become complicated. However, compatibility between two different versions of H.323 implementations is guaranteed. On the other hand, a new version of the SIP implementation does not need to support obsolete features from past versions. When an obsolete feature is requested from a server, the server, by default, ignores the request, informs the requestor of the unsupported feature and lets the requestor decide what to do. This way, a SIP implementation is typically cleaner while still maintaining some compatibility. Unlike H.323, different versions of SIP implementation may suffer from compatibility problems [10].

c. Interoperability

H.323 has higher interoperability than SIP. H.323 has well-defined implementation guidelines available to help improve interoperability among different H.323 vendors. Also, the H.32x family specifies standards to guarantee interoperability among circuit-switched networks such as ISDN, B-ISDN and GSTN. SIP is also highly interoperable with other protocols due to its flexibility and modular design. For example, SIP can be used in conjunction with H.323 where SIP provides the location service and H.323 performs the rest of the communication services. However, SIP is loosely defined and open to various interpretations. This may lead to potential interoperability issues. There is a growing effort focused on addressing interoperability issues in SIP [9].

3. Scalability

The scalability of the two protocols, or the ability to support small or large volume of data or users, is compared in below. The design, server components, and conference mechanisms of the protocols will be evaluated for scalability.

a. Protocol Design

H.323 was originally designed for local area networks. Addressing in a wide area network (WAN) and user location were not initial concerns for H.323. As networks employing H.323 have grown in size, H.323 has been augmented to address these issues. However, H.323 still has a scalability problem because its loop detection

algorithm using path values does not work well. SIP, on the other hand, was designed to support WAN addressing and user location. It uses a loop detection mechanism that is similar to the one employed by Border Gateway Protocol (BGP). Thus unlike H.323, the SIP loop detection algorithm scales well [10].

b. Servers

Scalability generally decreases if servers have to maintain state for all calls. Servers used in both protocols can be stateful or stateless. Endpoints in both protocols need to keep states in stateless call implementations. Endpoints as well as servers need to maintain states in stateful call implementation. The drawback of maintaining states is the large amount of memory and processing that is required. Most current H.323 gatekeeper implementations are designed to be call stateful whereas most SIP proxy implementations are designed to be call-stateless [12]. Therefore, SIP scales better in large networks.

c. Conferencing

H.323 relies on the Multipoint Control Unit (MCU) to manage signaling in multiparty conferences regardless of the number of participants. However, MCU can be a bottleneck in large conferences. SIP is more scalable with regard to conferencing because it employs a distributed control scheme [12].

4. Security

This section compares the security provided by H.323 and SIP based on [5]. More specifically, it examines the protection of authenticity, confidentiality, and integrity for both signaling and media data provided by H.323 and SIP.

a. H.323 Security - H.235

H.323's relies on H.235 to specify security standards. However, H.235 “does not mandate particular [security] features” [13]. To address interoperability among different H.235 vendors, H.235 defines security profiles corresponding different security levels. Table 5 summaries the different H.235 security profiles or annexes described in [5].³

³ H.235 Annex A (H.235 ASN.1), Annex B (H.323 Specific Topics), and Annex C (H.334 Specific Topics) are not listed in Table 4.

b. SIP Security

The SIP standard specifies several security features including HTTP Digest Authentication and S/MIME. The protocol does recommend other best security practices to address authentication, confidentiality, and integrity for both signaling and media data. Table 6 lists the existing SIP security features presented in [5].

c. Security Comparison

H.235, the security protocol for H.323, and SIP both have recommendations to protect the authenticity, confidentiality, and integrity for both signaling and media data. At the same time, ITU-T and IETF are making serious efforts to address security problems by continuously devising new security recommendations. Even though H.235 has effective security measures, H.323 does not mandate vendors to implement any of the H.235 security measures. According to an online website, not many H.323 products have support for H.235 and those that have “only use H.235 (baseline security) for the communication between gatekeeper and gateway and not for communication with the endpoint” [13]. SIP, on the contrary, has security mechanisms specified in the protocol implementation and is inherently more secure than H.323.

Annex	Name	Description	Protection
D	Baseline Security Profile	Protect messages with shared secrets and hashed values.	Authentication, Integrity
E	Signatures Security Profile	Protect messages with certificates and digital signatures.	Authentication, Integrity
	Voice Encryption Option	Encrypt voice streams. Can add on top of Annex D or E.	Confidentiality
F	Hybrid Security Profile	Establish shared secret using certificates and digital signatures. Use shared secret to protect messages.	Authentication, Integrity
G (draft)	SRTP & MIKEY Usage	Use MIKEY to manage keys in SRTP.	Authentication, Confidentiality, Replay
H (draft)	RAS Key Management	Protect keys established during gatekeeper discovery with PINs or passwords.	Confidentiality
I	H.235 Annex D for Direct Routed Scenarios	Enhances security provided by Annex D and F with option to use in direct routed calls.	Authentication, Integrity
J	N/A	Describes security for simple endpoint types.	Authentication, Integrity

Table 5. H.235 Security Profiles

Name	Description	Protection
HTTP Digest	Challenge and response scheme to protect signaling information. Endpoint sends to server an MD5 hash of the username, password, and nonce provided by server.	Authentication
S/MIME	Use S/MIME to provide protection to signaling information.	Authentication, Confidentiality, Integrity
RTP and SRTP	Encrypt RTP data defined in RFC 1889 or use SRTP to protect media data.	Confidentiality
SDP	Uses SDP to carry session keys for media data.	Confidentiality
TLS	SIP requires proxy and redirect servers and registrars and recommends endpoints to use TLS to protect signaling data.	Confidentiality, Integrity, Replay
IPsec	Use IPsec to provide security for signaling data at the network layer	Authentication, Confidentiality, Integrity
AIB (draft)	Use a digitally-signed message or message fragment to provide authentication	Authentication
Authenticated Identity Management (draft)	Recommends on how to correctly verify end users	Authentication
S/MIME AES Requirement (draft)	Use AES instead of DES or 3DES as a minimum requirement for encryption implementations of S/MIME in SIP	Confidentiality
Security Mechanism Agreement (draft)	Provide a method to negotiate which security mechanism to use between two endpoints	N/A
End-to-Middle, Middle-to-Middle, Middle-to-End Security (draft)	Describes security in different modes of communication (i.e. end-to-end, middle-to-middle, middle-to-end)	N/A

Table 6. SIP Security Features

5. Conclusion

Both protocols have strengths as well as weaknesses. SIP is more flexible and light-weight but less well-defined compared to H.323. H.323 has a detailed specification and offers higher interoperability but supports fewer applications. Nevertheless, H.323 and SIP are widely used in VoIP applications and both are undergoing more development to address their weaknesses. Neither of the two will become obsolete. Thus, interoperability between them will become necessary.

Nevertheless, SIP is simpler, more flexible, extensible, scalable, and can be more secure than H.323 based on the above comparison. SIP, the younger protocol of the two, is showing the potential to become a highly successful Internet protocol. Products based on SIP are becoming increasingly available for these reasons. For example, Microsoft has shifted H.323-based NetMeeting implementation to a SIP-based implementation in Windows XP. Furthermore, Microsoft also incorporates a SIP-like protocol stack in its .Net framework that can be used on desktops and mobile devices such as PDAs and smart phones [14].⁴ Further research on this young protocol is highly valuable to the community, as the number of applications supported by SIP is expected to grow. For this work, SIP has been selected for use in the experiments described in Chapter IV.

C. SUMMARY

H.323 and SIP provide similar VoIP services using different approaches. Thus they differ in simplicity and flexibility, extensibility, scalability, and security. Both protocols have advantages as well as disadvantages and they continue to be improved. For this project, SIP is chosen as the test protocol for research purposes.

⁴ NetMeeting is standard video-conferencing program included in Windows 2000 and XP.

IV. TESTING

This chapter describes the test methodology and test plan to verify the feasibility of SIP-based VoIP communications in different network architectures that included Network Address Translation (NAT) devices. An overview of the five tests and a brief summary of the findings are also presented. The testing described in this chapter is a preliminary step in integrating VoIP capabilities into the existing MYSEA architecture.

A. TEST METHODOLOGY

Testing is conducted on a dedicated testbed using an incremental approach. After each test, the result is thoroughly analyzed before proceeding to a more complicated test. The incremental testing approach is preferred in this study because it allows easy identification and debugging of problems that emerged during the tests.

A number of free tools are deployed in the testbed. In particular, SJPhone, a softphone developed by SJ Labs [15], is used to make and receive SIP-based VoIP calls. Ethereal [16], an open source packet capture tool, is used to capture packet exchanges during each VoIP session for post-testing analysis. *netfilter* and *iptables* [6], modules in Linux Operating System kernel, provide Network Address Translation and routing functions in the testbed. Finally, ZoneAlarm [17], a free software-based firewall, is used to block certain traffic during the tests.

A number of systems are used to model the different components that make up the MYSEA environment. For example, Windows laptops with SJPhone installed are in the testbed to represent the untrusted clients that sit behind the TPEs. Linux systems are used to perform NAT and/or routing functionalities. They simulate the TPEs in MYSEA.

This project focuses on testing the feasibility of making VoIP calls from the MLS LAN. Therefore, VoIP calls are always initiated from the clients located behind TPEs on the MLS LAN to the clients located on simulated single level networks. In terms of MYSEA, this is equivalent to allowing calls to be initiated from clients on the MLS LAN to clients on external networks.

B. TEST DESCRIPTION

The main objective of the tests described in the following subsections is to verify that VoIP conversations can be carried out in each network configuration. Procedures and results pertaining to each test can be found in Appendices B, C, D, E, F, and G. Note that private IP addresses assigned to network devices were used for demonstration purposes only. However, public devices such as public NATs and routers should use public IP addresses in practical scenarios.

1. Test 1: No NAT VoIP Configuration

The objective of this experiment is to observe the behavior of SJPhone in the simplest possible setup. The testbed consists of two directly connected VoIP-enabled clients as shown in Figure 6. In this scenario, Client B initiates a VoIP call to Client A. Test procedures and results are included in Appendix B.

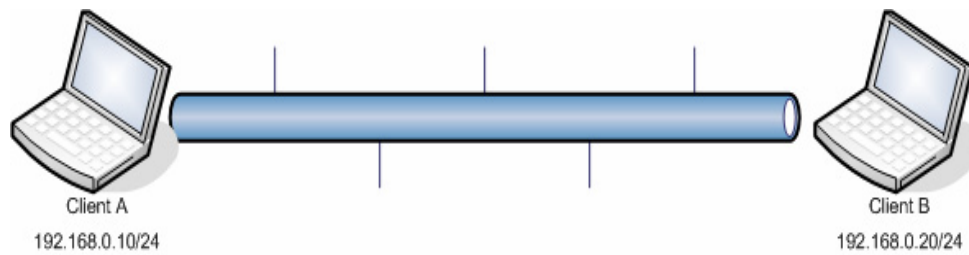


Figure 6. Test 1: Physical and Logical Network Topology

2. Test 2: Single NAT VoIP Configuration

The goal of this experiment is to confirm the feasibility of SIP-based VoIP calls when a NAT system is present. The testbed for this experiment consists of two VoIP-enabled clients and one NAT device as illustrated in Figure 7. The combination of the NAT device and Client B simulates the TPE-client pair in the MYSEA architecture. Client A is aware of Client B in this setup. Client B, on the other hand, is hidden behind a NAT device and is not visible to Client A. All packets exchanged between the two clients must traverse the NAT device that is configured with Source NAT and Destination NAT.

Two similar tests are conducted with this NAT configuration. The first test has a physical and logical network topology depicted in Figure 7. Even though the second test uses the same physical network topology as the first test, it has a logical topology

depicted in Figure 8. Since the NAT device is not configured to drop packets destined for private IP addresses, Client A can send RTP packets directly to the private IP address of Client B. This is exactly what Client A does based on the packet captures provided in Appendix C. In non-experimental scenarios, a firewall at the client is not necessary because packets destined for a private IP address will eventually be dropped as they traverse the networks. For demonstration purposes, a firewall is introduced at Client A to block packets initiated by Client A and destined for Client B. More information including the test procedures and results for both tests can be found in Appendix C.

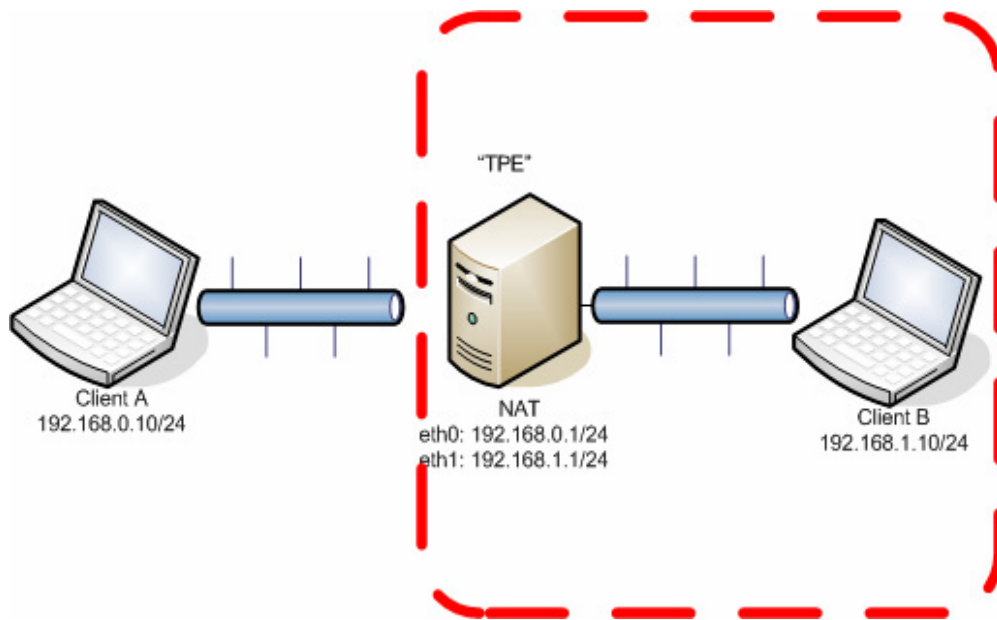


Figure 7. Test 2: Physical Network Topology (with and without firewall)

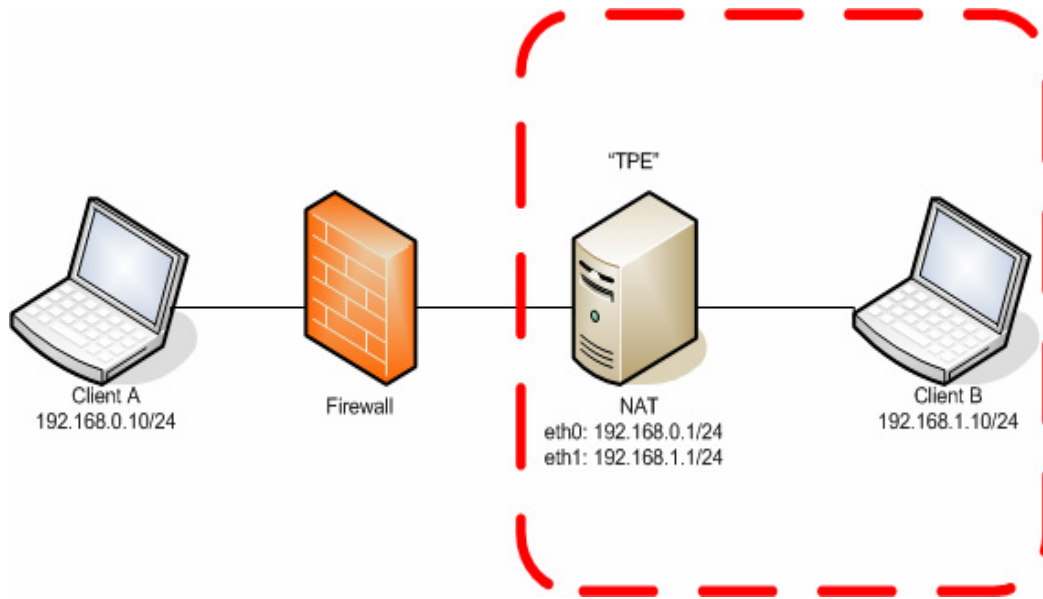


Figure 8. Test 2: Logical Network Topology (with firewall)

3. Test 3: Double NAT VoIP Configuration

The goal of this experiment is to confirm the feasibility of a SIP-based VoIP call using SJPhone when two NAT systems are present. In this test, a VoIP session between two clients have to traverse two different NAT devices as depicted in Figure 9 and Figure 10. Similar to the previous test, Client B and NAT 2 represent the client-TPE pair in the MYSEA architecture. NAT 1 simulates the NAT device located between the MYSEA network and the Internet whereas Client A acts as a VoIP-enabled client in the Internet. Both NAT devices are configured to perform Source NAT and Destination NAT. Test procedures and results are included in Appendix D.

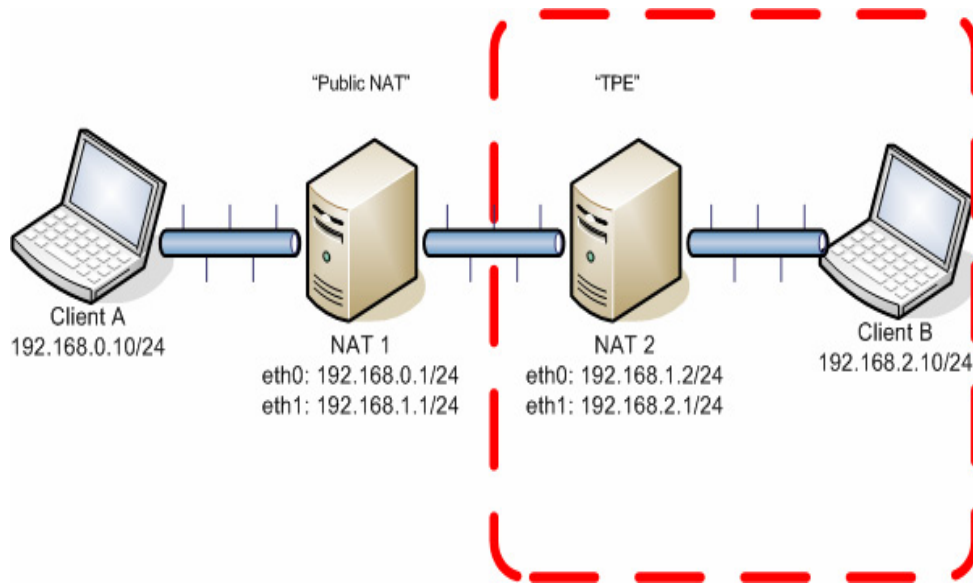


Figure 9. Test 3: Physical Network Topology

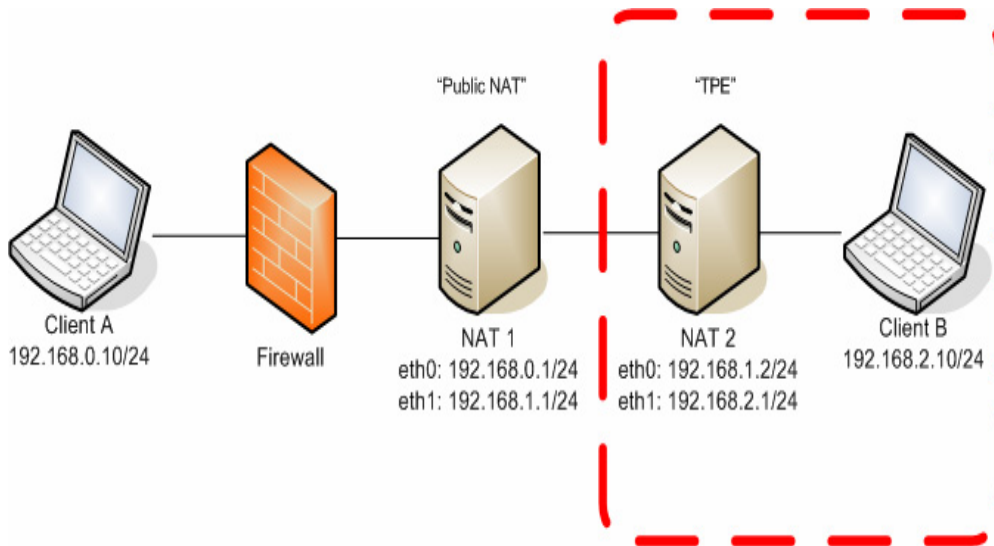


Figure 10. Test 3: Logical Network Topology

4. Test 4: Extended Double NAT VoIP Configuration

The purpose of this experiment is to confirm that two different VoIP sessions can take place at different times when the sessions have to traverse a common public NAT device. This and the next setup only work when the public NAT device implements a connection tracking mechanism that is similar to what *iptables* provides. The test is setup so that the public NAT device is not explicitly instructed to forward packets to any specific network. In other words, the public NAT only performs Source NAT but not

Destination NAT. A firewall is installed on Client A to prevent Client A from sending RTP packets directly to the private address of Client B. The firewall is necessary in this demonstration because NAT 1 and NAT 2 are not configured to drop packets destined for private IP addresses.

This test consists of three VoIP-enabled clients and three NAT devices as depicted in Figures 11 and 12. Two client-TPE pairs are simulated in this setup, Client B and NAT 2 being one pair and Client A and NAT 3 being the second pair. NAT 1 resembles the public NAT that is located between the MLS LAN and the Internet. NAT 1 is only configured with a SNAT rule whereas NAT 2 and NAT 3 are configured with both SNAT and DNAT rules. The test proceeds as follows: Client B initiates a call to Client A, terminates the call, and then Client C initiates a call to Client A. Procedures and results can be found in Appendix D.

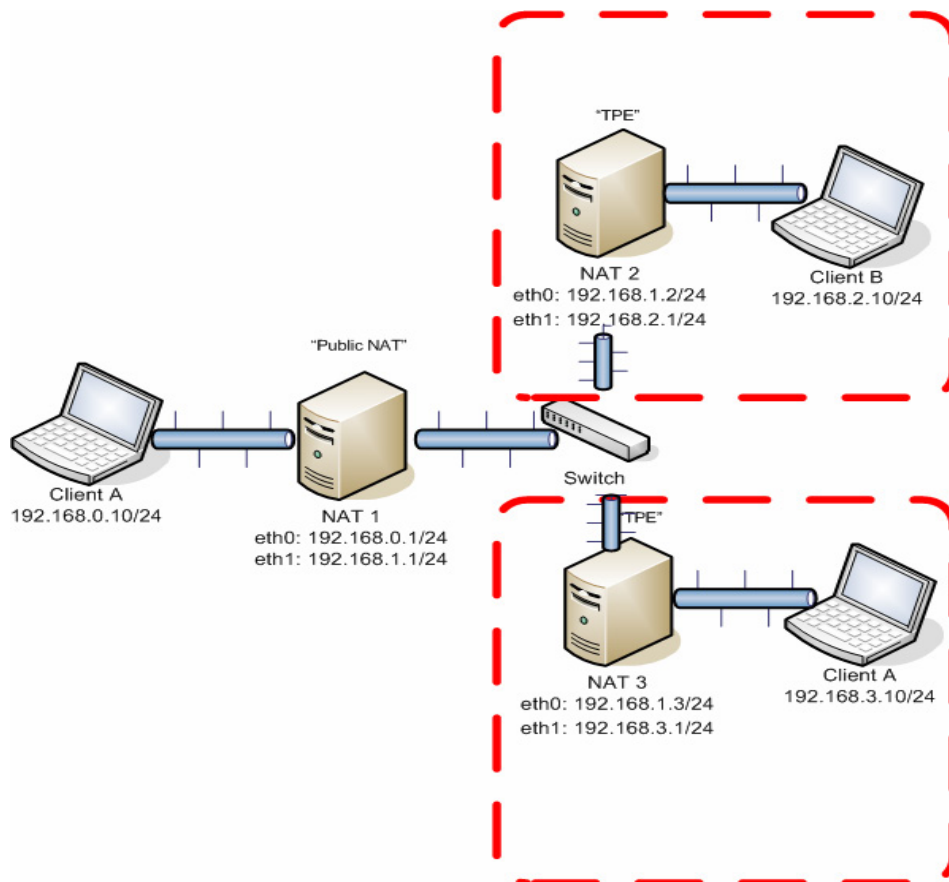


Figure 11. Test 4: Physical Network Topology

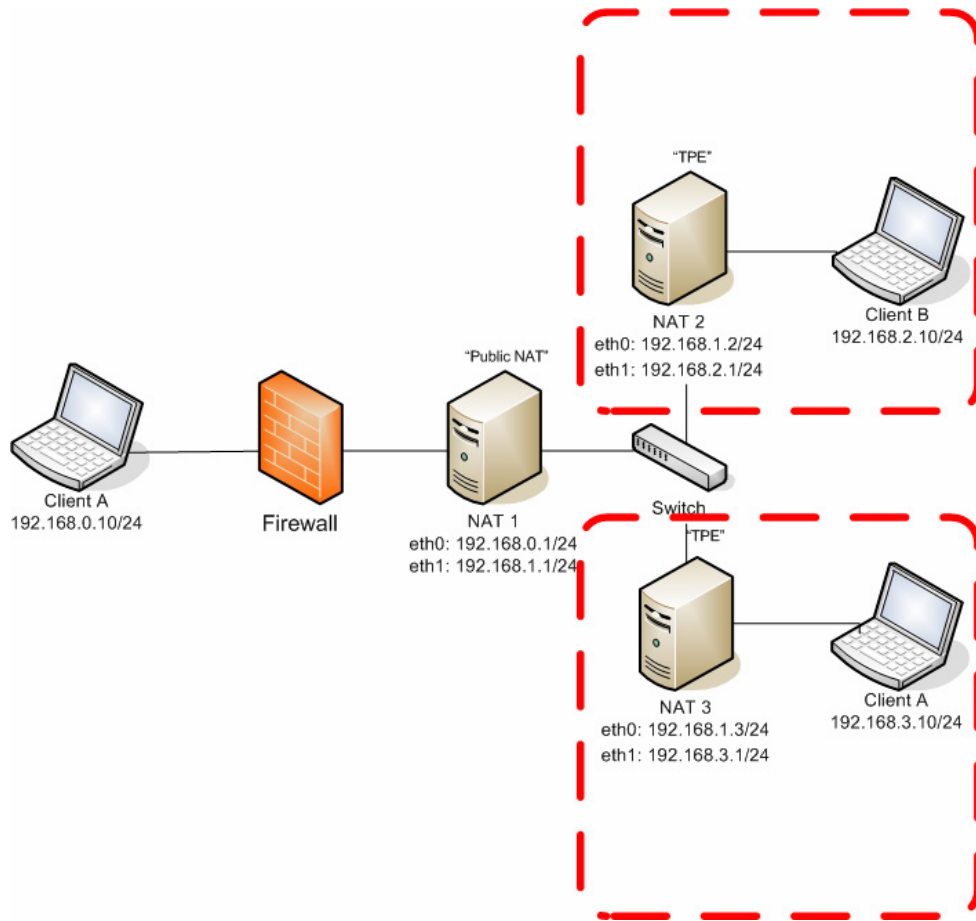


Figure 12. Test 4: Logical Network Topology

5. Test 5: Extended Double NAT VoIP Configuration with Simultaneous VoIP Sessions

The purpose of this test is to confirm the feasibility of two simultaneous VoIP sessions between two pairs of clients. The setup of this test, shown in Figures 13 and 14, closely resembles a simplified version of the MYSEA architecture. The IP addresses used for different components in this demonstration are the same as the ones used in the MYSEA testbed. This test consists of four VoIP-enabled clients, three NAT devices, and a router. The router is introduced here as a preparation for the next test. Refer to the next section for a description of the router. Similar to the previous test, two pairs of client-TPEs are simulated using Client C, NAT 2, Client D, and NAT 3. Furthermore, NAT 1 is not configured with a DNAT rule and two firewalls are installed on Client A and Client B

to block RTP packets destined to Client C and Client, respectively. In this scenario, Client C calls Client A and Client D calls Client B at the same time.

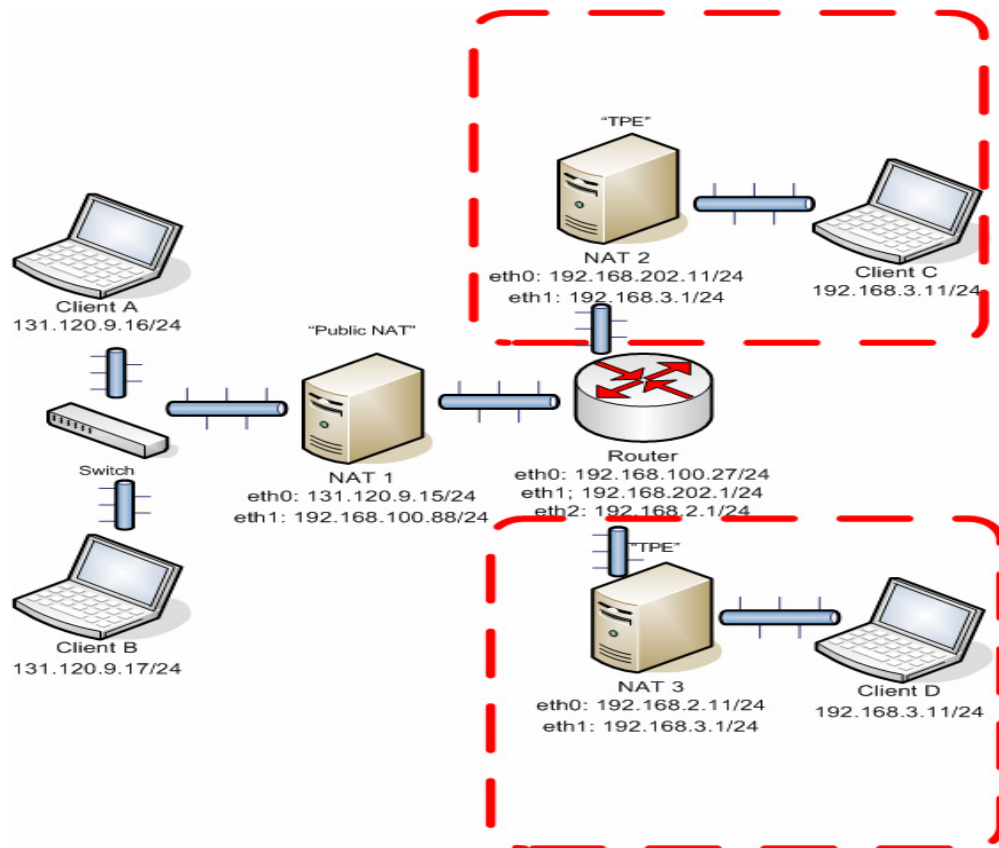


Figure 13. Test 5: Physical Network Topology

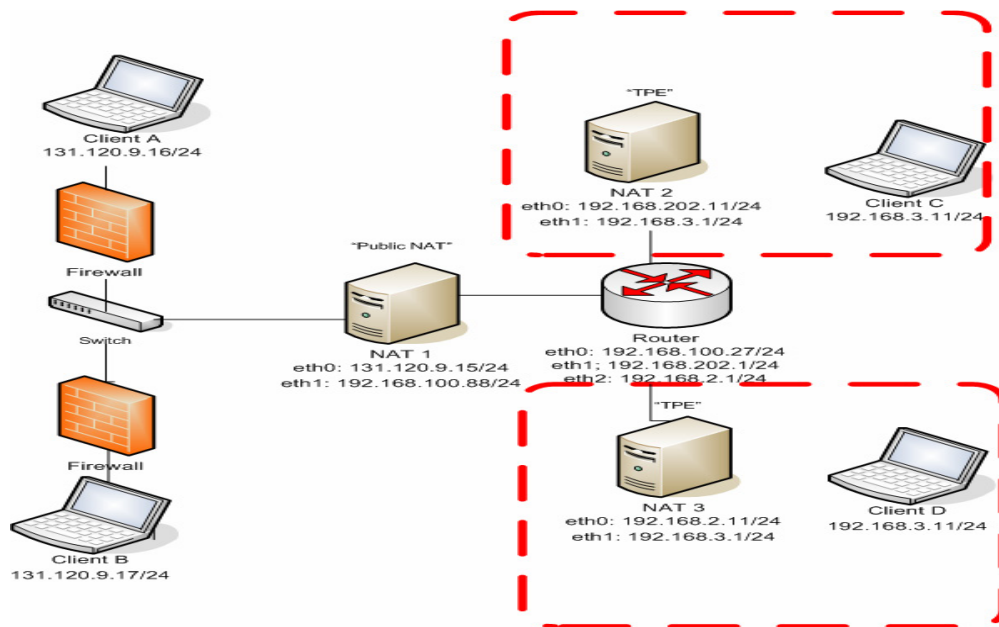


Figure 14. Test 5: Logical Network Topology

6. Test 6: MYSEA Configuration

The setup of this test, illustrated in figures 15 and 16, is an extension to the last one with the addition of a MLS server. The objective of this test is to verify that the MLS Server could support two simultaneous VoIP sessions between two pairs of clients. Since the MLS server does not perform routing in the testbed, a router is introduced to perform that function. The MLS server simply forwards the received packets to the correct network interfaces according to the network configurations. In this test, unexpected routing problems were encountered on the MLS server. Refer to the next section for a discussion of this test.

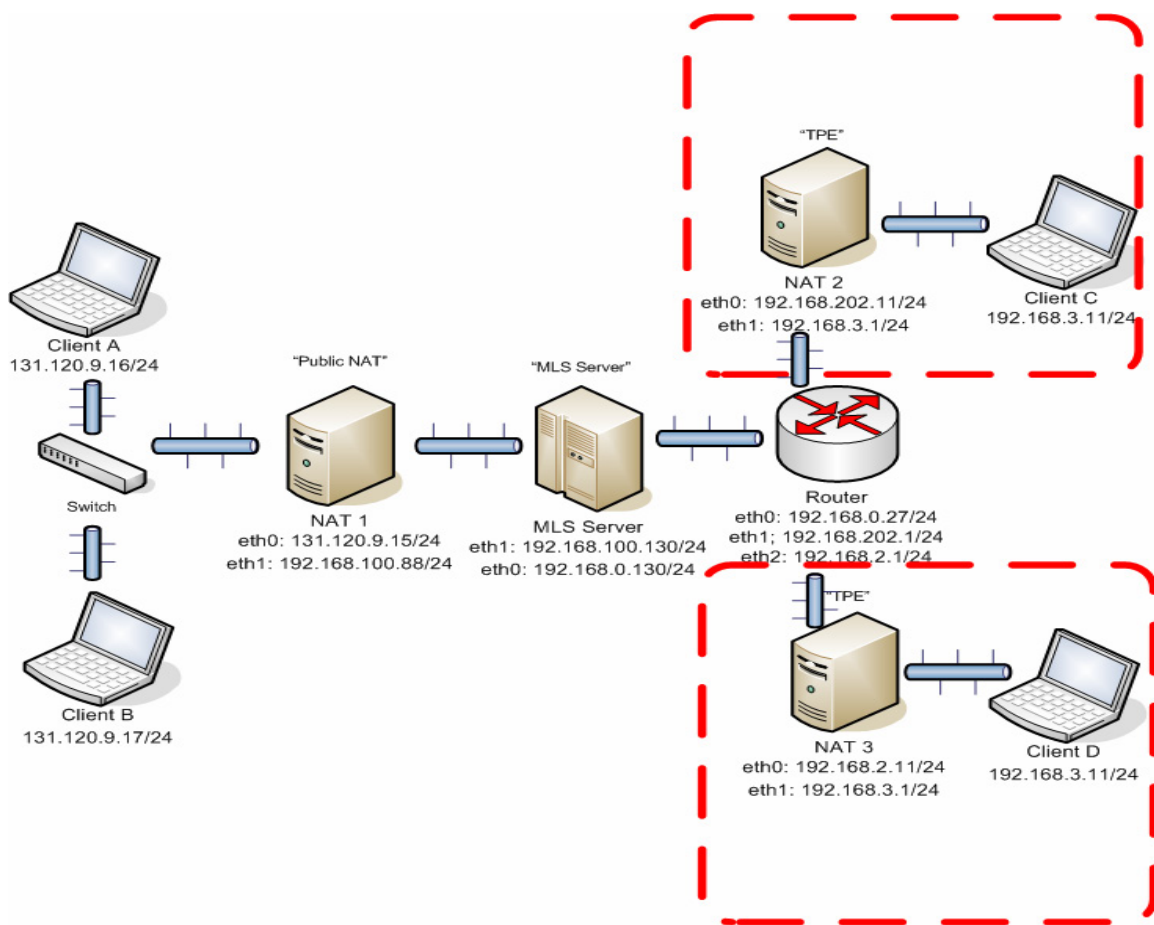


Figure 15. Test 6: Physical Network Topology

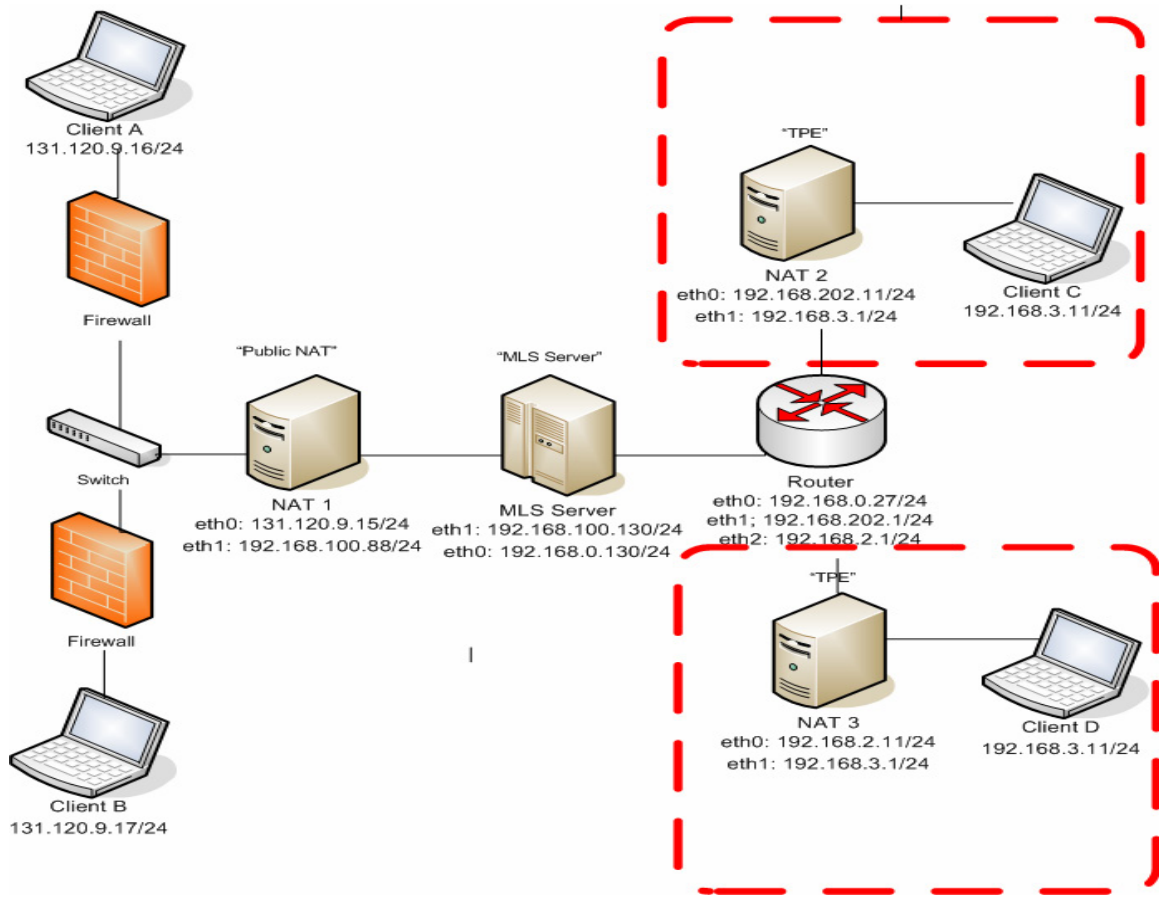


Figure 16. Test 6: Logical Network Topology

C. PROBLEMS ENCOUNTERED

Setting up and running the tests was fairly straightforward with the exception of the MYSEA Configuration. The major problem encountered that ultimately led to the failure of the MYSEA test was configuring the MLS server to perform proper routing. The MLS server, currently running XTS-400, has routing limitation such that only one static route can be configured for each network interface. An attempt to add a second route to the network interface X resulted in the following error message: “Route /dev/etherX already exists”. Since every packet has to traverse the server, this limitation prevents an incoming or outgoing packet arriving at the MLS server from routing to the other side of the network. According to the customer service response to our inquiry,, “there is a known restriction on [the XTS-400] configuration tool tcpip_edit (not the network stack) that there can only be one route per interface device...” [18]. Four test

scenarios were conducted to ensure that the MLS server, in fact, has routing limitation. Refer to Appendix G for details. In conclusion, the MYSEA Configuration test was not conducted successfully.

D. TEST RESULT

All the tests described in the previous section generate positive results, i.e. the results indicate that VoIP communications are possible in five of the six scenarios. However, it is important to recognize that the last test was unsuccessful not because of VoIP limitation or the network topology. Instead, the last test failed was due to configuration difficulties in the MLS server.

Several interesting findings are discovered from analyzing the packet captures. First, SJPhone will attempt to send RTP packets to the IP address indicated in the SDP data. If that fails, then SJPhone will resort to send subsequent RTP packets to the IP address it received RTP packets from. Second, the client that initiates the VoIP call is always the first to send out a RTP packet to the other party. Third, *iptables* has a connection tracking mechanism that allows it to associate incoming packets with previous outgoing packets and determine which VoIP session the incoming packets belong to. Since *iptables* creates an entry in its connection tracking table for the first RTP packet sent, subsequent incoming packets can be correctly forwarded to the correct next hop based on the information stored in the table. This mechanism plays a significant role in the success of tests 4 and 5 where the public NAT was not configured with a DNAT rule to forward incoming packets to any particular IP addresses. Refer to Appendices B through G for more details on the findings.

E. SUMMARY

The purposes and configurations of the experiments designed for this feasibility study are described in this Chapter. The results of the experiments are also briefly discussed. The results are optimistic and they indicate that the integration of VoIP into MYSEA is possible.

THIS PAGE INTENTIONALLY LEFT BLANK

V. FUTURE WORK AND CONCLUSIONS

A. FUTURE WORK

The test results described in Chapter IV and various appendices suggest that VoIP capabilities can be potentially integrated into the existing MYSEA architecture with little effort. However, further research in the following areas is required.

1. Routing in MLS Server

Every MLS client communication is mediated by the MLS server that runs on XTS-400, a high-assurance Unix-like system. Unfortunately, the MLS server has routing limitation such that only one static route can be configured for each network interface. The ability to configure more than one route for each interface in the MYSEA server is necessary for VoIP packets to route between clients on the MLS network and on external networks. Therefore, further study of the routing configurations or capabilities is required to allow proper routing of VoIP packets.

2. VoIP Conversations Initiated from the Internet

This research study is primarily concerned with testing the scenarios in which VoIP conversations are initiated from the MLS LAN. The ability for externally initiated VoIP conversation is also desirable. Currently, a client on an external network only knows the IP address of the public NAT device and there exists no way of distinguishing calls intended for different clients on the MLS LAN. In order for an internal client to receive an external call, three features may have to be implemented. First, each internal client must own a unique SIP address that is publicly known. This allows an external client to direct call to a specific internal client. Second, a server must exist to translate a SIP address to the corresponding internal client IP address. Third, softphones with reconfigurable RTP port, such as the SipXphone, are needed for each internal client. Each client needs to use a different RTP port for sending and receiving RTP packets and the public NAT must be configured to perform port forwarding. This allows NAT to forward RTP packets to the correct client according to the destination port. Research on how to implement this scheme is highly recommended.

B. CONCLUSIONS

The tests conducted in this research study were generally successful. Furthermore, the test results indicate that VoIP conversations, at least in the scenario we studied, between internal and external clients are possible even when various NAT devices are present. It is important to recognize that the success of Test 4 and Test 5 was dependent on the connection tracking mechanism in *iptables*. NAT devices without connection tracking mechanisms were not tested in this project. Thus, it is unknown whether the tests will work if those devices are used instead. In conclusion, VoIP capabilities may be integrated into the existing MYSEA architecture.

APPENDIX A. A SURVEY OF VOIP HARDPHONES AND SOFTPHONES

A. HARDPHONES

The following table is a survey of some VoIP hardphones. Each hardphone is listed with information including its manufacture (Brand/Company), phone type (category and Sub-Category), the VoIP protocol (VoIP Protocol), and Wi-Fi protocol (Wi-Fi Protocol) it supports.

Name	Brand/Company	Category	Sub-Category	VoIP Protocol	Wi-Fi Protocol
ACT P202	Advantage Century Telecomm Phones	WLAN/WiFi Phone	Mobile IP Phone	SIP	802.11b
Senao SI-7800H	Senao	WLAN/WiFi Phone	Mobile IP Phone	SIP	802.11b
F1000	UTStarcom	WLAN/WiFi Phone	Mobile IP Phone	SIP	802.11b
WLAN600	BCM	WLAN/WiFi Phone	Cordless Handset	SIP	802.11b
Cisco 7920	Cisco	WLAN/WiFi Phone	Cordless Handset	Skynny	802.11b
WirelessIP5000	Hitachi Cable	WLAN/WiFi Phone	Cordless Handset	SIP	802.11b
ZyXEL P2000W	ZyXEL	WLAN/WiFi Phone	Cordless Handset	SIP	802.11b
Vesta 100	Arkon Networks	WLAN/WiFi Phone	Cordless Handset	N/A	802.11b/g
Motorola CN620	Motorola	GSM/WiFi Phone	Handset	SIP	802.11a/b/g
Nobstrand 200	Arkon Networks	WLAN/WiFi Phone	Handset	SIP	802.11b/g
Stonehenge WP150	Molmstone	WLAN/WiFi Phone	Wireless Phone	SIP	802.11b/g

Table 7. Summary of Hardphones

B. SOFTPHONES

The following table is a survey of some VoIP softphones. Each softphone is listed with information including what Operating System(s) and VoIP protocol(s) (VoIP Protocol) it supports and whether it is a commercial or an open source product.

Name	Pocket PC 2003	MAC OS X	Linux	WinZK	WinXP	VoIP Protocol	Commercial	Open Source
X-Life/eyeBeam	Y	Y		Y	Y	SIP	Y	
X-Pro	Y	Y		Y	Y	SIP	Y	
eyeP Phone Lite				Y	Y	SIP		Y
SIPPS						SIP	Y	
Ubiquity User Agent				Y	Y	SIP		Y
EZ-Phone				Y	Y	SIP	Y	
MySIP				Y	Y	SIP	Y	
SIPphone	Y	Y	Y	Y	Y	SIP and H.323	Y	
Linphone			Y			SIP		Y
Kphone			Y			N/A		Y
Vovida			Y			SIP		Y
Siphon			Y			SIP		Y
slipXphone			Y	Y	Y	SIP		Y
Shbom		Y		Y	Y	SIP		Y
Corfedd SIP-UA			Y			SIP		Y
SFLphone		Y		Y	Y	SIP		Y
Asterisk			Y			SIP and H.323		Y
Empower Pro Internet Phone	Y	Y				N/A	Y	
Skype		Y	Y	Y	Y	SIP	Y	

Table 8. Summary of Softphones

APPENDIX B. TEST 1: NO NAT VOIP DEMONSTRATION USING SJPHONE

The instructions in this appendix describe how to setup and demonstrate a SIP-based VoIP communication between two directly connected SIP-enabled clients using SJPhone. Figure 6 illustrates the physical network as well as the logical topology for this demonstration. A VoIP session is initiated from Client B to Client A. Packet captures from both clients are included at the end of this appendix along with an analysis.

A. Network Topology

Refer to Figure 6 for the physical and logical network topology.

B. Equipment Requirements

B.1. Clients A and B

B.1.1. Windows XP Operating System

B.1.2. Sound card

B.1.3. SJPhone v.1.60

B.1.4. Ethereal

B.2. Additional Equipment

B.2.1. Cross-over cable to implement the network architecture Figure 6

B.2.2. Microphones as audio input devices for clients A and B

C. Installation and Configuration

C.1. Client A

IP Address: 192.168.0.10

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.0.20

C.2. Client B

IP Address: 192.168.0.20

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.0.10

C.3. SJPhone Installation and Configuration

C.3.1. Client A and Client B

- C.3.1.1. Download the Windows version of SJPhone v.1.60 from SJ Labs
 - C.3.1.2. Install SJPhone v.1.60
 - C.3.1.3. Launch SJPhone
 - C.3.1.4. Right-click on SJPhone
 - C.3.1.5. Go to Services
 - C.3.1.6. Select PC-to-PC (SIP)
- C.4. **Ethereal Installation and Configuration**
 - C.4.1. **Clients A and B**
 - C.4.1.1. Download the latest Windows version of Ethereal
 - C.4.1.2. Install Ethereal
- D. **Preparation and Testing**
 - D.1. Adjust volume on both clients accordingly
 - D.2. Plug microphones into both clients
 - D.3. On Client A,
 - D.3.1. Launch Ethereal
 - D.3.2. Go to the **Capture** menu
 - D.3.3. Go to **Interfaces**
 - D.3.4. Click on **Capture 192.168.0.10**
 - D.4. On Client B,
 - D.4.1. Launch Ethereal
 - D.4.2. Go to the **Capture** menu
 - D.4.3. Go to **Interfaces**
 - D.4.4. Click on **Capture 192.168.0.20**
 - D.4.5. Call Client A by dialing 192.168.0.10 in SJPhone
 - D.5. On Client A,
 - D.5.1. Select **Accept** in the pop-up dialog box when SJPhone rings
 - D.5.2. Clients A and B may engage in a VoIP conversation at this point
 - D.5.3. Click on the **Hang-Up** bottom on either SJPhone to terminate the call
when finished
 - D.6. On Clients A and B,
 - D.6.1. Stop packet captures by selecting **Stop** on Ethereal

E. Packet Captures

E.1. Client A

Figure 17 is a snapshot of the packets captured on Client A.

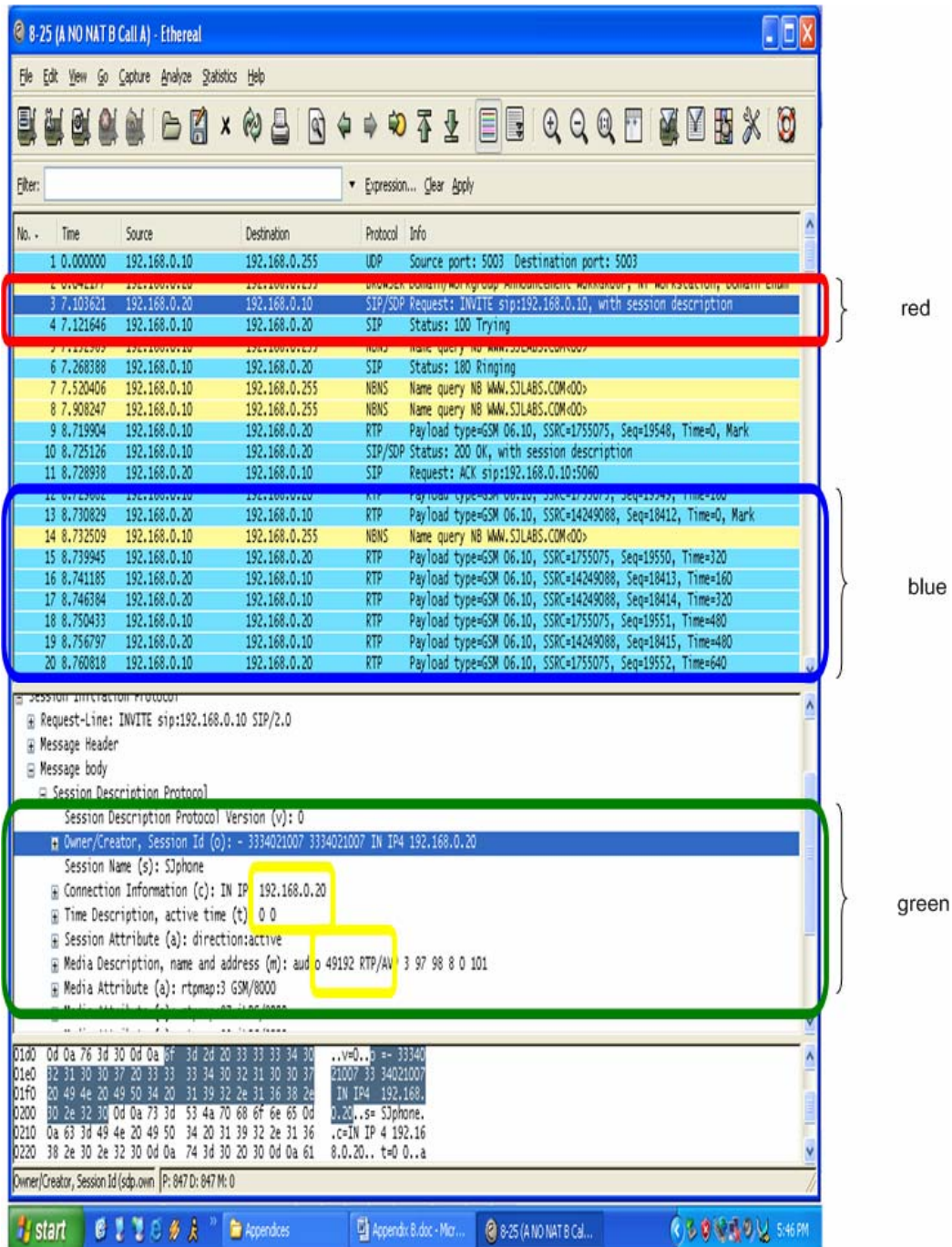


Figure 17. Test 1: Packet Capture on Client A

E.2. Client B

Figure 18 is a snapshot of the packets captured on Client B.

The screenshot shows a Wireshark capture of network traffic on Client B. The main packet list table is as follows:

No.	Time	Source	Destination	Protocol	Info
3	7.367234	192.168.0.20	192.168.0.10	SIP/SDP	Request: INVITE sip:192.168.0.10, with session description
4	7.390962	192.168.0.10	192.168.0.20	SIP	Status: 100 Trying
5	7.412628	192.168.0.10	192.168.0.255	NBNS	Name query NB WWW.SJLABS.COM<00>
6	7.672186	192.168.0.10	192.168.0.20	SIP	Status: 180 Ringing
7	8.112268	192.168.0.10	192.168.0.255	NBNS	Name query NB WWW.SJLABS.COM<00>
8	8.503570	192.168.0.10	192.168.0.255	NBNS	Name query NB WWW.SJLABS.COM<00>
9	9.333330	192.168.0.10	192.168.0.20	KIP	Payload type=GSM 06.10, SSRC=1735075, Seq=19548, Time=0, Mark
10	9.333833	192.168.0.10	192.168.0.20	SIP/SDP	Status: 200 OK, with session description
11	9.334465	192.168.0.20	192.168.0.10	SIP	Request: ACK sip:192.168.0.10:5060
12	9.335518	192.168.0.10	192.168.0.20	KIP	Payload type=GSM 06.10, SSRC=1735075, Seq=19549, Time=160
13	9.336387	192.168.0.20	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=14249088, Seq=18412, Time=0, Mark
14	9.338157	192.168.0.10	192.168.0.255	NBNS	Name query NB WWW.SJLABS.COM<00>
15	9.345580	192.168.0.10	192.168.0.20	RTP	Payload type=GSM 06.10, SSRC=1735075, Seq=19550, Time=320
16	9.346743	192.168.0.20	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=14249088, Seq=18413, Time=160
17	9.351943	192.168.0.20	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=14249088, Seq=18414, Time=320
18	9.356063	192.168.0.10	192.168.0.20	RTP	Payload type=GSM 06.10, SSRC=1735075, Seq=19551, Time=480
19	9.362359	192.168.0.20	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=14249088, Seq=18415, Time=480
20	9.366449	192.168.0.10	192.168.0.20	RTP	Payload type=GSM 06.10, SSRC=1735075, Seq=19552, Time=640
21	9.372781	192.168.0.20	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=14249088, Seq=18416, Time=640
22	9.376729	192.168.0.10	192.168.0.20	RTP	Payload type=GSM 06.10, SSRC=1735075, Seq=19553, Time=800

The details pane for packet 11 (SIP Request: ACK) is expanded, showing the Session Description Protocol (SDP) section. The SDP details are as follows:

- Session Description Protocol Version (v): 0
- Owner/Creator, Session Id (o): 3334042171.3334042171 IN IP4 192.168.0.10
- Session Name (s): SJphone
- Connection Information (c): IN IP4 192.168.0.1
- Time Description, active time (t): 0 0
- Session Attribute (a): direction:active
- Media Description, name and address (m): audio 49170 RTP/VP 3 101
- Media Attribute (a): rtptime=1 GSM/ROO

The orange box highlights packets 10 and 11 in the packet list. The purple box highlights the SDP details for packet 11. The yellow box highlights the Session Name (s) and Connection Information (c) fields in the SDP details.

Figure 18. Test 1: Packet Capture on Client B

E.3. Analysis

The packet captures indicate that as soon as Client B initiated a call to Client A, Client B sent out an “INVITE” message from 192.168.0.20:5060 to Client A at 192.168.0.10:5060 (red outline in Figure 17). The “INVITE” message had embedded SDP information to inform Client A that Client B will send and receive RTP voice packets at 192.168.0.20 on port 49192 (green outline in Figure 17). Client A acknowledged the invitation by sending Client B a “200 OK” message (orange outline in Figure 18) with embedded SDP information indicating that it will send and receive RTP voice packets at 192.168.0.10 on port 49170 (purple outline in Figure 18). Subsequent voice exchanges between the two clients were achieved via 192.168.0.20: 49192 on Client B and 192.16.0.10: 49170 on Client A (blue outline in Figure 17).

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. TEST 2: SINGLE NAT VOIP DEMONSTRATION USING SJPHONE

The instructions contained in this appendix describe how to setup and demonstrate a SIP-based VoIP communication between two SIP-enabled clients via a Network Address Translation (NAT) device. In this setup, Clients A and B belong to different networks and Client B is located behind a NAT device. The NAT device is configured to act as a router and modify the destination or source IP address of all packets that traverse it. In this scenario, Client B initiates a VoIP call to Client A.

The demonstration consists of two parts. They are very similar in nature except that a firewall is introduced in the second part. Packet captures and an analysis are included for each part.

A. Without Firewall

A.1. Network Topology

Refer to Figure 7 for the physical and logical network topology.

A.2. Equipment Requirements

A.2.1. Client A and Client B

A.2.1.1. Windows XP Operating System

A.2.1.2. Sound Card

A.2.1.3. SJPhone v.1.60

A.2.1.4. Ethereal

A.2.1.5. ZoneAlarm (Client A only)

A.2.2. NAT

A.2.2.1. Linux Operating System (Fedora Core 4)

A.2.2.2. netfilter and iptables

A.2.2.3. Ethereal

A.2.2.4. Two network cards

A.2.3. Additional Equipment

A.2.3.1. Cross-over cables to implement the network architecture illustrated in Figure 7

A.2.3.2. Microphones as audio input devices for Client A and Client B

A.2.4. Installation and Configuration

A.2.4.1. Client A

IP Address: 192.168.0.10

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.0.1

A.2.4.2. Client B

IP Address: 192.168.0.20

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.1.1

A.2.4.3. NAT

A.2.4.3.1. Configure eth0 by editing /etc/sysconfig/network-scripts/ifcfg-eth0:

DEVICE=eth0

BOOTPROTO=NONE

IPADDR=192.168.0.1

NETMASK=255.255.255.0

A.2.4.3.2. Activate eth0 by running:

ifup eth0

A.2.4.3.3. Configure eth1 by editing /etc/sysconfig/network-scripts/ifcfg-eth1:

DEVICE=eth1

BOOTPROTO=NONE

IPADDR=192.168.1.1

NETMASK=255.255.255.0

A.2.4.3.4. Activate eth1 by running:

`ifup eth1`

A.2.4.3.5. Enable IP Forwarding by running:

`echo 1 > /proc/sys/net/ipv4/ip_forward`

A.2.4.3.6. Flush any existing firewall and NAT rules by running:

`iptables -F`

`iptables -t nat -F`

A.2.4.3.7. Configure NAT rules by running:

`iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 192.168.0.1`

`iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 192.168.1.10`

A.2.4.4. SJPhone Installation and Configuration

A.2.4.4.1. Client A and Client B

A.2.4.4.1.1. Download the Windows version of SJPhone v.160
from SJLabs

A.2.4.4.1.2. Install SJPhone v.160

A.2.4.4.1.3. Launch SJPhone

A.2.4.4.1.4. Right-click on SJPhone

A.2.4.4.1.5. Go to **Services**

A.2.4.4.1.6. Select **PC-to-PC (SIP)**

A.2.4.5. Ethereal Installation and Configuration

A.2.4.5.1. Client A and Client B

A.2.4.5.1.1. Download the latest Windows version of Ethereal

A.2.4.5.1.2. Install Ethereal

A.2.4.5.2. NAT

A.2.4.5.2.1. Install Ethereal if it is not already installed:

A.2.4.5.2.1.1. Go to the **Desktop** menu

A.2.4.5.2.1.2. Go to **System Settings**

A.2.4.5.2.1.3. Go to **Add/Remove Applications**

A.2.4.5.2.1.4. Click on **Details** under **System Tools**

- A.2.4.5.2.1.5. Find and then check **ethereal-gnome**
- A.2.4.5.2.1.6. Click on **Close**
- A.2.4.5.2.1.7. Click on **Update**
- A.2.4.5.2.1.8. Put in the correct Fedora Core 4 CDs when prompted

A.3. Preparation and Testing

- A.3.1. Adjust volume on both clients accordingly
- A.3.2. Plug microphones into both clients
- A.3.3. On client A,
 - A.3.3.1. Launch **Ethereal**
 - A.3.3.2. Go to the **Capture** menu
 - A.3.3.3. Go to **Interfaces**
 - A.3.3.4. Click on **Capture 192.168.0.10**
- A.3.4. On client B,
 - A.3.4.1. Launch **Ethereal**
 - A.3.4.2. Go to the **Capture** menu
 - A.3.4.3. Go to **Interfaces**
 - A.3.4.4. Click on **Capture 192.168.1.10**
- A.3.5. On NAT,
 - A.3.5.1. Launch one instance of **Ethereal**
 - A.3.5.2. Go to the **Capture** menu
 - A.3.5.3. Go to **Interfaces**
 - A.3.5.4. Click on **Capture Eth0**
 - A.3.5.5. Launch another instance of **Ethereal**
 - A.3.5.6. Go to the **Capture** menu
 - A.3.5.7. Go to **Interfaces**
 - A.3.5.8. Click on **Capture Eth1**
- A.3.6. On Client B,
 - A.3.6.1. Call A by dialing 192.168.0.10 in **SJPhone**
- A.3.7. On Client A,
 - A.3.7.1. Select **Accept** in the pop-up dialog box when **SJPhone** rings

A.3.8. Clients A and B may engage in a VoIP conversation at this point

A.3.9. Click on the Hang-Up bottom on either SJPhone to terminate the call
when finished

A.3.10. On Client A, Client B, and NAT,

A.3.10.1. Stop packet captures by selecting **Stop** on Ethereal

A.4. Packet Captures

A.4.1. Client A

Figure 19 is a snapshot of the packets captured on Client A.

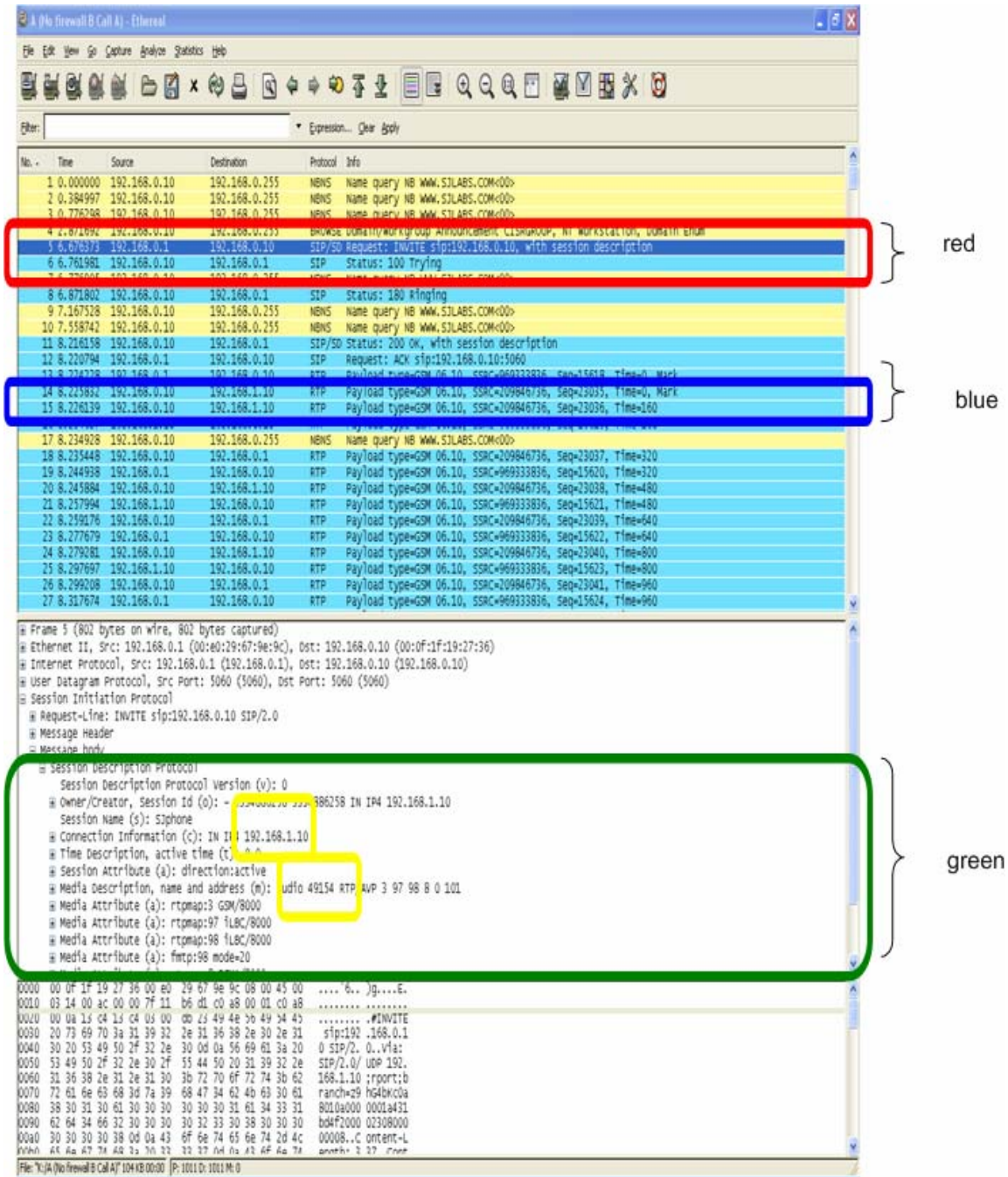


Figure 19. Test 2: Packet Capture on Client A (without firewall)

A.4.2. Client B

Figure 20 is a snapshot of the packets captured on Client B.



Figure 20. Test 2: Packet Capture on Client B (without firewall)

A.4.3. NAT eth0

Figure 21 is a snapshot of the packets captured on the first interface (eth0) of the NAT device.

The screenshot displays the Wireshark network protocol analyzer interface. At the top, the title bar reads "Wireshark 1.10.4 [60 bytes on wire (60 bytes captured) on interface eth0]". The main window is divided into three panes:

- Packet List:** Shows a list of captured packets. The first packet (No. 61) is a DNS query from 192.168.0.10 to 192.168.0.1. Subsequent packets show responses and further queries.
- Packet Details:** Provides a hierarchical view of the selected packet's structure. For the first packet, it shows the Ethernet II header, Internet Protocol Version 4 header, and the DNS message structure (Standard query query).
- Packet Bytes:** Displays the raw data of the selected packet in hexadecimal and ASCII format.

The packet list shows a series of DNS queries and responses. The first packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The second packet is a response from 192.168.0.1 to 192.168.0.10. The third packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The fourth packet is a response from 192.168.0.1 to 192.168.0.10. The fifth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The sixth packet is a response from 192.168.0.1 to 192.168.0.10. The seventh packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The eighth packet is a response from 192.168.0.1 to 192.168.0.10. The ninth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The tenth packet is a response from 192.168.0.1 to 192.168.0.10. The eleventh packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The twelfth packet is a response from 192.168.0.1 to 192.168.0.10. The thirteenth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The fourteenth packet is a response from 192.168.0.1 to 192.168.0.10. The fifteenth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The sixteenth packet is a response from 192.168.0.1 to 192.168.0.10. The seventeenth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The eighteenth packet is a response from 192.168.0.1 to 192.168.0.10. The nineteenth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The twentieth packet is a response from 192.168.0.1 to 192.168.0.10. The twenty-first packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The twenty-second packet is a response from 192.168.0.1 to 192.168.0.10. The twenty-third packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The twenty-fourth packet is a response from 192.168.0.1 to 192.168.0.10. The twenty-fifth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The twenty-sixth packet is a response from 192.168.0.1 to 192.168.0.10. The twenty-seventh packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The twenty-eighth packet is a response from 192.168.0.1 to 192.168.0.10. The twenty-ninth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The thirtieth packet is a response from 192.168.0.1 to 192.168.0.10. The thirty-first packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The thirty-second packet is a response from 192.168.0.1 to 192.168.0.10. The thirty-third packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The thirty-fourth packet is a response from 192.168.0.1 to 192.168.0.10. The thirty-fifth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The thirty-sixth packet is a response from 192.168.0.1 to 192.168.0.10. The thirty-seventh packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The thirty-eighth packet is a response from 192.168.0.1 to 192.168.0.10. The thirty-ninth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The fortieth packet is a response from 192.168.0.1 to 192.168.0.10. The forty-first packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The forty-second packet is a response from 192.168.0.1 to 192.168.0.10. The forty-third packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The forty-fourth packet is a response from 192.168.0.1 to 192.168.0.10. The forty-fifth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The forty-sixth packet is a response from 192.168.0.1 to 192.168.0.10. The forty-seventh packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The forty-eighth packet is a response from 192.168.0.1 to 192.168.0.10. The forty-ninth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The fiftieth packet is a response from 192.168.0.1 to 192.168.0.10. The fifty-first packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The fifty-second packet is a response from 192.168.0.1 to 192.168.0.10. The fifty-third packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The fifty-fourth packet is a response from 192.168.0.1 to 192.168.0.10. The fifty-fifth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The fifty-sixth packet is a response from 192.168.0.1 to 192.168.0.10. The fifty-seventh packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The fifty-eighth packet is a response from 192.168.0.1 to 192.168.0.10. The fifty-ninth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The sixtieth packet is a response from 192.168.0.1 to 192.168.0.10. The sixty-first packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The sixty-second packet is a response from 192.168.0.1 to 192.168.0.10. The sixty-third packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The sixty-fourth packet is a response from 192.168.0.1 to 192.168.0.10. The sixty-fifth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The sixty-sixth packet is a response from 192.168.0.1 to 192.168.0.10. The sixty-seventh packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The sixty-eighth packet is a response from 192.168.0.1 to 192.168.0.10. The sixty-ninth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The seventieth packet is a response from 192.168.0.1 to 192.168.0.10. The seventy-first packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The seventy-second packet is a response from 192.168.0.1 to 192.168.0.10. The seventy-third packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The seventy-fourth packet is a response from 192.168.0.1 to 192.168.0.10. The seventy-fifth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The seventy-sixth packet is a response from 192.168.0.1 to 192.168.0.10. The seventy-seventh packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The seventy-eighth packet is a response from 192.168.0.1 to 192.168.0.10. The seventy-ninth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The eightieth packet is a response from 192.168.0.1 to 192.168.0.10. The eighty-first packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The eighty-second packet is a response from 192.168.0.1 to 192.168.0.10. The eighty-third packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The eighty-fourth packet is a response from 192.168.0.1 to 192.168.0.10. The eighty-fifth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The eighty-sixth packet is a response from 192.168.0.1 to 192.168.0.10. The eighty-seventh packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The eighty-eighth packet is a response from 192.168.0.1 to 192.168.0.10. The eighty-ninth packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The ninetieth packet is a response from 192.168.0.1 to 192.168.0.10. The ninety-first packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The ninety-second packet is a response from 192.168.0.1 to 192.168.0.10. The ninety-third packet is a query for "www.32labs.com" from 192.168.0.10 to 192.168.0.1. The ninety-fourth packet is a response from 192.168.0.1 to 192.168.0.10. The ninety-fifth packet is a query for "www.32labs.com" from 192.1

Figure 21. Test 2: Packet Capture on eth0 of NAT (without firewall)

A.4.4. NAT eth1

Figure 22 is a snapshot of the packets captured on the second interface (eth1) of the NAT device.

The screenshot shows a Wireshark packet capture on the 'eth1' interface of a NAT device. The packet list pane displays 27 captured packets. The first few packets include ARP requests and replies, ICMP echo requests and replies, and a UDP packet. The majority of the packets (from packet 6 to 27) are RTP packets, which are part of a media stream. The packet details pane shows the structure of the first packet (Frame 1), identifying it as an ARP request. The packet bytes pane shows the raw hexadecimal and ASCII data of the first packet.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.1	Broadcast	ARP	who has 192.168.1.10? (eth1:192.168.1.1)
2	0.000104	192.168.1.10	192.168.1.1	ARP	192.168.1.10 is at 00:0f:17:18:d7:c5
3	0.000116	192.168.0.10	192.168.1.10	ICMP	Echo (ping) request
4	0.000218	192.168.1.10	192.168.0.10	ICMP	Echo (ping) reply
5	46.590860	192.168.1.10	192.168.1.255	UDP	Source port: 5003 destination port: 5003
6	111.24322	192.168.1.10	192.168.1.255	BROADCAST	Domain/Workgroup Announcement WORKGROUP, NT workstation, domain Enum
7	116.15616	192.168.1.10	192.168.0.10	SDP/SD	Request: INVITE sip:192.168.0.10, with session description
8	116.32072	192.168.0.10	192.168.1.10	SDP	Status: 100 trying
9	116.53170	192.168.0.10	192.168.1.10	SDP	Status: 180 Ringing
10	119.10790	192.168.0.10	192.168.1.10	SDP/SD	Status: 200 OK, with session description
11	119.11614	192.168.1.10	192.168.0.10	SDP	Request: ACK sip:192.168.0.10:5060
12	119.12296	192.168.1.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=98933836, Seq=15618, Time=0, Mark
13	119.12624	192.168.0.10	192.168.1.10	RTP	Payload type=GSM 06.10, SSRC=209846736, Seq=20035, Time=0, Mark
14	119.12682	192.168.0.10	192.168.1.10	RTP	Payload type=GSM 06.10, SSRC=209846736, Seq=20036, Time=160
15	119.14388	192.168.1.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=98933836, Seq=15619, Time=160
16	119.14467	192.168.0.10	192.168.1.10	RTP	Payload type=GSM 06.10, SSRC=209846736, Seq=20037, Time=320
17	119.16267	192.168.1.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=98933836, Seq=15620, Time=320
18	119.16467	192.168.0.10	192.168.1.10	RTP	Payload type=GSM 06.10, SSRC=209846736, Seq=20038, Time=480
19	119.18299	192.168.1.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=98933836, Seq=15622, Time=480
20	119.18434	192.168.0.10	192.168.1.10	RTP	Payload type=GSM 06.10, SSRC=209846736, Seq=20039, Time=640
21	119.20267	192.168.1.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=98933836, Seq=15623, Time=640
22	119.20444	192.168.0.10	192.168.1.10	RTP	Payload type=GSM 06.10, SSRC=209846736, Seq=20040, Time=800
23	119.22269	192.168.1.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=98933836, Seq=15625, Time=800
24	119.22437	192.168.0.10	192.168.1.10	RTP	Payload type=GSM 06.10, SSRC=209846736, Seq=20042, Time=960
25	119.24265	192.168.1.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=98933836, Seq=15624, Time=960
26	119.24431	192.168.0.10	192.168.1.10	RTP	Payload type=GSM 06.10, SSRC=209846736, Seq=20042, Time=1120
27	119.26267	192.168.1.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=98933836, Seq=15625, Time=1120

Frame 1 (42 bytes on wire, 42 bytes captured)
Ethernet II, Src: 192.168.1.1 (00-01-01-01-01-01), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)

0000 ff ff ff ff ff ff 02 08 00 00 00 00 00 00 00 00
0010 00 00 00 00 00 00 02 08 00 00 00 00 00 00 00 00
0020 00 00 00 00 00 00 00 c0 a8 02 0a
.....

Figure 22. Test 2: Packet Capture on eth1 of NAT (without firewall)

A.4.5. Analysis

Since all packets exchanged between Clients A and B are processed by the NAT rules, understanding those rules is essential when analyzing the traffic flow captured by Ethereal. The SNAT rule `iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 192.168.0.1` instructed the NAT device to modify the source IP address of all outgoing packets to 192.168.0.1 before routing them. Thus, all the packets received by Client A appeared to come from the NAT device (see packet capture for Client A). The DNAT rule `iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 192.168.1.10` instructed the NAT device to change the destination IP address of all incoming packets to 192.168.1.10 before routing those packets. This operation allowed packets sent to the NAT to be routed to Client B.

The process of establishing a connection between Clients A and B in this test was very similar to the one described in Appendix B. First, Client B sent out an “INVITE” message from 192.168.1.10:5060 to 192.168.0.10:5060 on Client A (red outline in Figure 19). The “INVITE” message had embedded SDP information to inform Client A that Client B will send and receive RTP voice packets at 192.168.1.10 on port 49154 (green outline in Figure 19). Client A acknowledged the invitation by sending Client A an “200 OK” message with embedded SDP information indicating that it will send and receive RTP packets at 192.168.0.10 on port 49152 (orange outline in Figure 20).

Subsequent voice communication between the two clients was sent to the IP addresses and ports specified in SDP. In other words, Client B sent RTP packets directly to Client A at 192.168.0.10 and Client A sent RTP packets directly to Client B at 192.168.1.10 (blue outline in Figure 19). The former was legitimate because Client A was publicly reachable. But the latter was only possible in our setup since neither Client A nor the NAT device was configured to drop packets destined for private IP addresses. In this case, the NAT device simply forwarded the RTP packets to client B (see Figures 21 and 22). To simulate a more realistic network configuration, a firewall was needed to drop packets sent by Client A and destined for Client B.

B. With Firewall

B.1. Network Topology

Refer to Figure 7 and Figure 8 for the physical and logical network topology.

B.2. Preparation and Testing (in addition to all steps described in Section A)

B.2.1. On Client A,

B.2.1.1. Download the ZoneAlarm from Zone Labs

B.2.1.2. Install ZoneAlarm

B.2.1.3. When ZoneAlarm is being run for the first time, it will ask the user to choose between Basic ZoneAlarm or the trial version of ZoneAlarm Pro, select the trial version of ZoneAlarm

B.2.1.4. When asked to select a security level for the detected network, select Allow into Trusted Zone

B.2.1.5. Configure firewall rule in ZoneAlarm:

B.2.1.5.1. Go to Firewall menu on the left panel

B.2.1.5.2. Click on the Expert tab

B.2.1.5.3. Click on Add

B.2.1.5.4. Type in a name for the firewall rule in the Name textbox

B.2.1.5.5. Under Action, select Block

B.2.1.5.6. Under Destination,

B.2.1.5.6.1. Select Modify

B.2.1.5.6.2. Select Add Location

B.2.1.5.6.3. Select IP Address

B.2.1.5.6.4. Type in a description in the Description textbox

B.2.1.5.6.5. Type 192.168.1.10 in the IP Address textbox

B.2.1.5.6.6. Click OK

B.2.1.5.6.7. Click OK

B.2.1.5.6.8. Click Apply

B.2.2. Run test as described in Section A

B.3. Packet Captures

B.3.1. Client A

Figure 23 is a snapshot of the packets captured on Client A.



Figure 23. Test 2: Packet Capture on Client A (with firewall)

B.3.2. Client B

Figure 24 is a snapshot of the packets captured on Client B.



Figure 24. Test 2: Packet Capture on Client B (with firewall)

Figure 25 is a snapshot of the packets captured on the first interface (eth0) of the NAT device.



B.3.4. NAT eth1

Figure 26 is a snapshot of the packets captured on the second interface (eth1) of the NAT device.

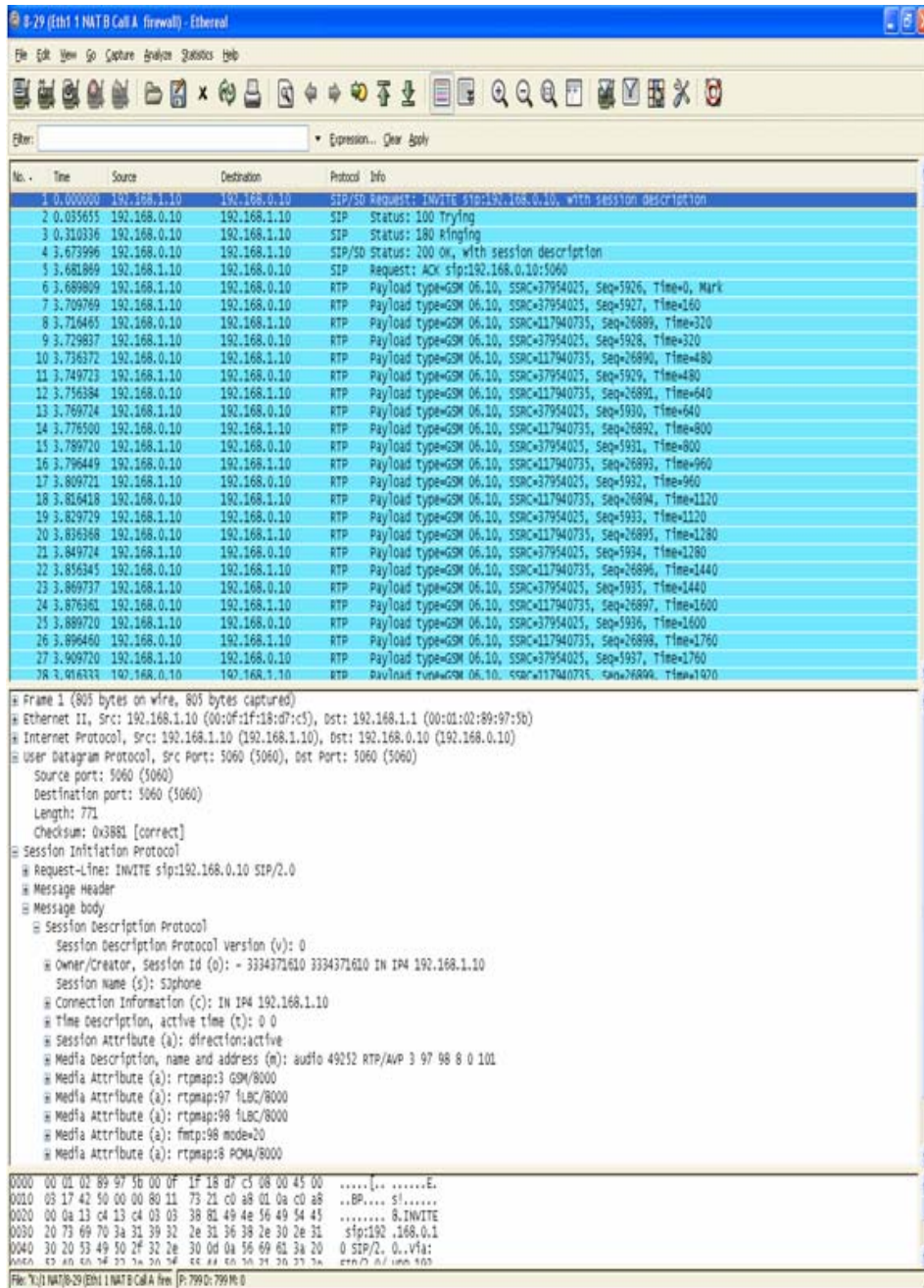


Figure 26. Test 2: Packet Capture on eth1 of NAT (with firewall)

B.3.5. Analysis

The signaling part of the call was processed as usual with the exchanges of “INVITE” (red outline in Figure 23) and “200 OK” (orange outline in Figure 24) messages between the two clients. Client B still told Client A to send RTP media packets to its private IP address or 192.168.1.10 (green outline in Figure 23). However, Client A can no longer send RTP packets directly to Client B at its private address because ZoneAlarm was configured to block those packets. The ZoneAlarm log files were examined to confirm that packets destined for 192.168.1.10 were, in fact, dropped. The packet captures on Client A indicate that when Client A failed to send RTP packets to the IP address of Client B, it tried to send subsequent RTP packets to the IP address from which it received RTP packets (due to the SNAT rule). In this case, Client A sent the RTP packets to the public IP address NAT or 192.168.0.1 (blue outline in Figure 23). When NAT received the packets, it modified the destination address in the packet header according to the configured DNAT rule. In other words, NAT changed the destination address from its own public IP address (192.168.0.1) to the private IP address of Client B (192.168.1.10) before forwarding the packets (dark red outline in Figure 24). Figures 25 and 26 confirm that DNAT and SNAT were done correctly.

APPENDIX D. TEST 3: DOUBLE NAT VOIP DEMONSTRATION USING SJPHONE

The instructions contained in this appendix describe how to setup and demonstrate a SIP-based VoIP communication between two SIP-enabled clients via two Network Address Translation (NAT) devices. In this setup, Client B is located behind two NATs. Each NAT is configured to act as a router and modifies the destination or source IP address of all packets that traverses it. Packet captures from both clients are included at the end of this appendix along with an analysis.

A. Network Topology

Refer to Figures 9 and Figure 10 for the physical and logical network topology.

B. Equipment Requirements

B.1. Clients A and B

B.1.1. Windows XP Operating System

B.1.2. Sound card

B.1.3. SJPhone v.1.60

B.1.4. Ethereal

B.1.5. ZoneAlarm (Client A only)

B.2. NAT 1 and NAT 2

B.2.1. Linux Operating System (Fedora Core 4)

B.2.2. netfilter and iptables

B.2.3. Ethereal

B.2.4. Two network cards

B.3. Additional equipment

B.3.1. Cross-over cables to implement the network architecture illustrated in
Figure 9

B.3.2. Microphones as audio input devices for clients A and B

C. Installation and Configuration

C.1. Client A

IP Address: 192.168.0.10

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.0.1

C.2. Client B

IP Address: 192.168.2.10

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.2.1

C.3. NAT 1

C.3.1. Configure eth0 by editing /etc/sysconfig/network-scripts/ifcfg-eth0:

DEVICE=eth0

BOOTPROTO=NONE

IPADDR=192.168.0.1

NETMASK=255.255.255.0

C.3.2. Activate eth0 by running:

ifup eth0

C.3.3. Configure eth1 by editing /etc/sysconfig/network-scripts/ifcfg-eth1:

DEVICE=eth1

BOOTPROTO=NONE

IPADDR=192.168.1.1

NETMASK=255.255.255.0

C.3.4. Activate eth1 by running:

ifup eth1

C.3.5. Enable IP Forwarding by running:

echo 1 > /proc/sys/net/ipv4/ip_forward

C.3.6. Flush any existing firewall and NAT rules by running:

iptables -F

iptables -t nat -F

C.3.7. Configure NAT rules by running:

iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 192.168.0.1

iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 192.168.1.2

C.4. NAT 2

C.4.1. Configure eth0 by editing /etc/sysconfig/network-scripts/ifcfg-eth0:


```
DEVICE=eth0
BOOTPROTO=NONE
IPADDR=192.168.1.2
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
```

C.4.2. Activate eth0 by running:

```
ifup eth0
```

C.4.3. Configure eth1 by editing /etc/sysconfig/network-scripts/ifcfg-eth1:

```
DEVICE=eth1
BOOTPROTO=NONE
IPADDR=192.168.2.1
NETMASK=255.255.255.0
```

C.4.4. Activate eth1 by running:

```
ifup eth1
```

C.4.5. Enable IP Forwarding by running:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

C.4.6. Flush any existing firewall and NAT rules by running:

```
iptables -F
iptables -t nat -F
```

C.4.7. Configure NAT rules by running:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 192.168.1.2
iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 192.168.2.10
```

C.5. SJPhone Installation and Configuration

C.5.1. Clients A and B

- C.5.1.1. Download the Windows version of SJPhone v.1.60 from SJ Labs
- C.5.1.2. Install SJPhone v.1.60
- C.5.1.3. Launch SJPhone
- C.5.1.4. Right-click on SJPhone
- C.5.1.5. Right-click
- C.5.1.6. Go to Services
- C.5.1.7. Select PC-to-PC (SIP)

C.6. Ethereal Installation and Configuration

C.6.1. Clients A and B

C.6.1.1. Download the Windows version of Ethereal v.0.10.12

C.6.1.2. Install Ethereal v.0.10.12 by following on-screen instructions

C.6.2. NAT 1 and NAT 2

C.6.2.1.1. Install Ethereal if it is not already installed

C.6.2.1.1.1. Go to the **Desktop** menu

C.6.2.1.1.2. Go to **System Settings**

C.6.2.1.1.3. Go to **Add/Remove Applications**

C.6.2.1.1.4. Click on **Details** under **System Tools**

C.6.2.1.1.5. Find and then check **ethereal-gnome**

C.6.2.1.1.6. Click on **Close**

C.6.2.1.1.7. Click on **Update**

C.6.2.1.1.8. Put in the correct Fedora Core 4 CDs when
prompted

C.7. ZoneAlarm Installation and Configuration

C.7.1. On client A,

C.7.1.1. Download the free ZoneAlarm from Zone Labs

C.7.1.2. Install ZoneAlarm by following on-screen instructions

C.7.1.3. When ZoneAlarm is being run for the first time, it will ask the user
to choose between Basic ZoneAlarm or trial version of ZoneAlarm Pro,
select the trial version of ZoneAlarm

C.7.1.4. Answer on-screen questions

C.7.1.5. When asked to select a security level for the detected network,
select **Allow into Trusted Zone**

C.7.1.6. Configure firewall rule in ZoneAlarm:

C.7.1.6.1. Go to **Firewall** menu on the left panel

C.7.1.6.2. Click on the **Expert** tab

C.7.1.6.3. Click on **Add**

C.7.1.6.4. Type in a name for the firewall rule in the **Name** textbox

C.7.1.6.5. Under **Action**, select **Block**

C.7.1.6.6. Under **Destination**,

- C.7.1.6.6.1. Select **Modify**
- C.7.1.6.6.2. Select **Add Location**
- C.7.1.6.6.3. Select **IP Address**
- C.7.1.6.6.4. Type in a description in the **Description** textbox
- C.7.1.6.6.5. Type 192.168.2.10 in the **IP Address** textbox
- C.7.1.6.6.6. Click **OK**
- C.7.1.6.6.7. Click **OK**
- C.7.1.6.6.8. Click **Apply**

D. Preparation and Testing

D.1. Adjust volume on both clients accordingly

D.2. Plug microphones into both clients

D.3. On Client A,

D.3.1. Launch **Ethereal**

D.3.2. Go to the **Capture** menu

D.3.3. Go to **Interfaces**

D.3.4. Click on **Capture 192.168.0.10**

D.4. On Client B,

D.4.1. Launch **Ethereal**

D.4.2. Go to the **Capture** menu

D.4.3. Go to **Interfaces**

D.4.4. Click on **Capture 192.168.2.10**

D.5. On NAT 1 (Ethereal not installed),

D.5.1. Launch one terminal and run:

```
tcpdump -n -i eth0
```

D.5.2. Launch another terminal and run:

```
tcpdump -n -i eth1
```

D.6. On NAT 2,

D.6.1. Launch one instance of **Ethereal**

D.6.1.1. Go to the **Capture** menu

D.6.1.2. Go to **Interfaces**

- D.6.1.3. Click on **Capture Eth0**
- D.6.2. Launch another instance of Ethereal
 - D.6.2.1. Go to the **Capture** menu
 - D.6.2.2. Go to **Interfaces**
 - D.6.2.3. Click on **Capture Eth1**
- D.7. On Client B,
 - D.7.1. Call A by dialing 192.168.0.10 in SJPhone
- D.8. On Client A,
 - D.8.1. Select **Accept** in the pop-up dialog box when SJPhone rings
 - D.8.2. Clients A and B may engage in a VoIP conversation at this point.
 - D.8.3. Click on the **Hang-Up** bottom on either SJPhone to terminate call when finished
 - D.8.4. On NAT 1,
 - D.8.4.1. Stop tcpdump packet captures by pressing **Control-C** on the terminals
 - D.8.5. On Client A, Client B and NAT 2,
 - D.8.5.1. Stop packet captures by selecting **Stop** on Ethereal

E. Packet Captures

E.1. Client A

The following is a snapshot of the packets captured on Client A.

The screenshot shows a Wireshark packet capture on Client A. The top pane displays a list of captured packets. A red box highlights packet 1, which is a SIP/SDP Request: INVITE sip:192.168.0.10, with session description. A blue box highlights packet 11, which is an RTP payload type=GSM 06.10. A green box highlights the details of packet 1, showing the Session Initiation Protocol (SIP) message structure.

Red Box (Packet 1):

- 1 0.000000 192.168.0.1 192.168.0.10 SIP/SD Request: INVITE sip:192.168.0.10, with session description

Blue Box (Packet 11):

- 11 1.035593 192.168.0.1 192.168.0.10 RTP Payload type=GSM 06.10, SSRC=516040833, Seq=12039, Time=0, Mark

Green Box (Packet 1 Details):

- Frame 1 (805 bytes on wire (805 bytes captured))
- Ethernet II, Src: 00:4c:69:6e:75:79 (00:4c:69:6e:75:79), Dst: 192.168.0.10 (00:0f:1f:19:27:36)
- Internet Protocol, Src: 192.168.0.1 (192.168.0.1), Dst: 192.168.0.10 (192.168.0.10)
- User Datagram Protocol, Src Port: 5060 (5060), Dst Port: 5060 (5060)
- Source port: 5060 (5060)
- Destination port: 5060 (5060)
- Checksum: 0x8aaf [correct]
- Session Initiation Protocol
- Request-Line: INVITE sip:192.168.0.10 SIP/2.0
- Message Header
- Message body
- Session Description Protocol
- Session Description Protocol version (v): 0
- Owner/Creator, Session Id (o): - 3334024859 3334024859 IN IP4 192.168.2.10
- Session Name (s): S3phone
- Connection Information (c): IN 1 4 192.168.2.10
- Time description, active time (t): 0 0
- Session Attribute (a): direction:active
- Media Description, name and address (m): audio 49204 RTP/AVP 3 97 98 8 0 101
- Media Attribute (a): rtptime:3 GSM/8000
- Media Attribute (a): rtptime:97 fLBC/8000
- Media Attribute (a): rtptime:98 fLBC/8000
- Media Attribute (a): rtptime:8 PCMA/8000

Figure 27. Test 3: Packet Capture on Client A

E.2. Client B

The following is a snapshot of the packets captured on Client B.

The screenshot shows a Wireshark packet capture of a SIP/SDP INVITE and RTP stream. The packet list shows the following details:

- Packet 7: SIP/SDP Status: 200 OK, with session description. This packet is highlighted with an orange box.
- Packet 8: RTP, payload type GSM 06.10, SSRC=516040833, Seq=12039, Time=0. This packet is also highlighted with an orange box.

The packet details pane shows the following information for the selected packet (Frame 7):

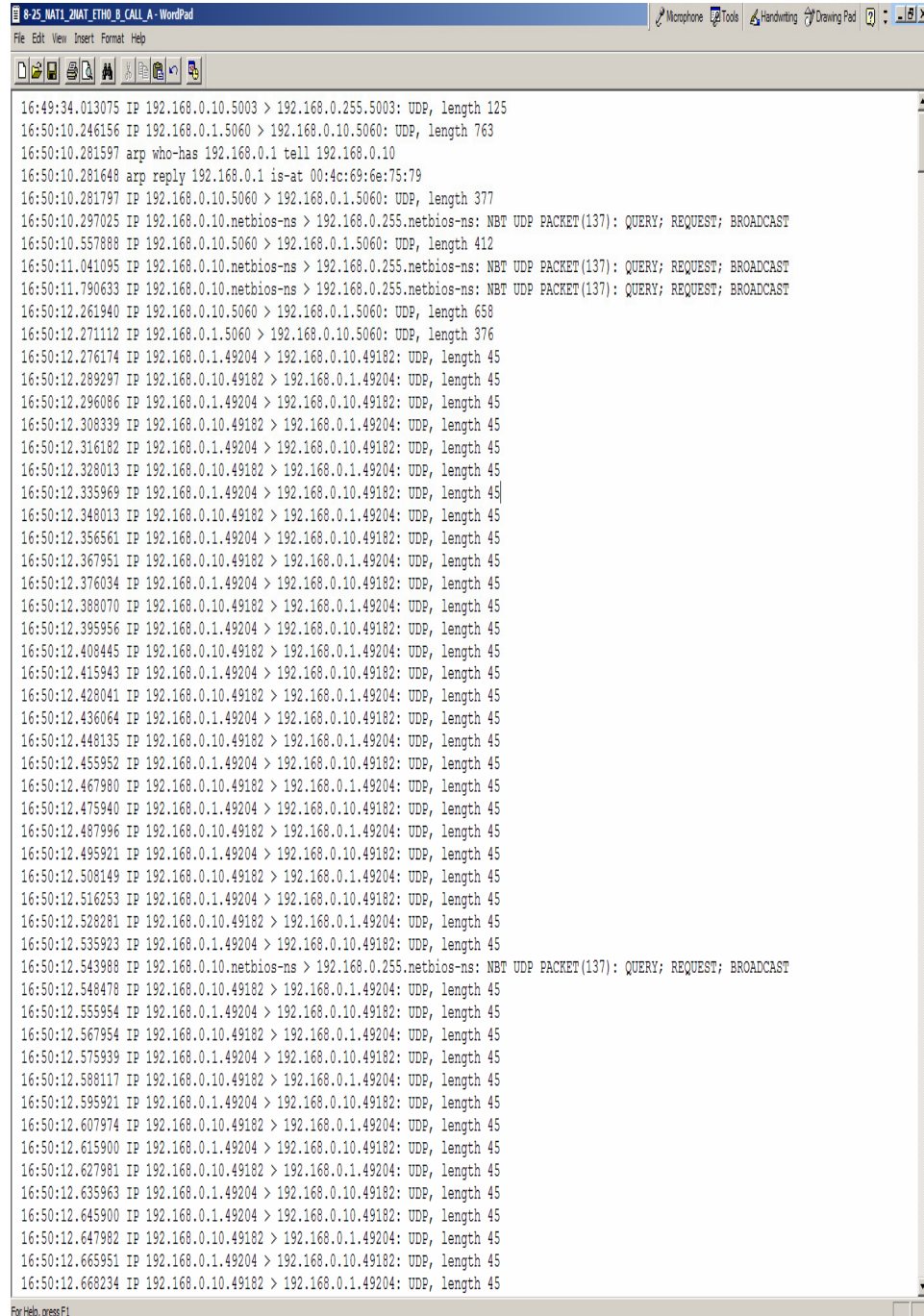
- Ethernet II, Src: 192.168.1.1 (00:01:02:89:97:5b), Dst: 192.168.1.10 (00:0f:1f:18:d7:c5)
- Internet Protocol, Src: 192.168.0.10 (192.168.0.10), Dst: 192.168.2.10 (192.168.2.10)
- User Datagram Protocol, Src Port: 5060 (5060), Dst Port: 5060 (5060)
- Source port: 5060 (5060)
- Destination port: 5060 (5060)
- Length: 666
- Session Initiation Protocol
 - Status-Line: SIP/2.0 200 OK
 - Message Header
 - Message body
 - Session Description Protocol
 - Session Description Protocol version (v): 0
 - Owner/Creator, Session Id (o): - 3334046023 3334046023 IN IP4 192.168.0.10
 - Session Name (s): S2phone
 - Connection Information (c): IN IP4 192.168.0.10
 - Time Description, active time (t): 0 0
 - Session Attribute (a): direction:active
 - Media Description, name and address (m): udfo 49182 RTP RVP 3 101
 - Media Attribute (a): rtmap:3 GSM/8000
 - Media Attribute (a): rtmap:101 telephone-event/8000
 - Media Attribute (a): fmtp:101 0-11,16

The packet bytes pane shows the raw data of the packet, including the Ethernet II header, Internet Protocol header, User Datagram Protocol header, and the SIP/SDP message body.

Figure 28. Test 3: Packet Capture on Client B

E.3. NAT 1 Eth0

The following is a snapshot of the packets captured on the first interface (eth0) of NAT 1.



```
16:49:34.013075 IP 192.168.0.10.5003 > 192.168.0.255.5003: UDP, length 125
16:50:10.246156 IP 192.168.0.1.5060 > 192.168.0.10.5060: UDP, length 763
16:50:10.281597 arp who-has 192.168.0.1 tell 192.168.0.10
16:50:10.281648 arp reply 192.168.0.1 is-at 00:4c:69:6e:75:79
16:50:10.281797 IP 192.168.0.10.5060 > 192.168.0.1.5060: UDP, length 377
16:50:10.297025 IP 192.168.0.10.netbios-ns > 192.168.0.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:50:10.557888 IP 192.168.0.10.5060 > 192.168.0.1.5060: UDP, length 412
16:50:11.041095 IP 192.168.0.10.netbios-ns > 192.168.0.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:50:11.790633 IP 192.168.0.10.netbios-ns > 192.168.0.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:50:12.261940 IP 192.168.0.10.5060 > 192.168.0.1.5060: UDP, length 658
16:50:12.271112 IP 192.168.0.1.5060 > 192.168.0.10.5060: UDP, length 376
16:50:12.276174 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.289297 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.296086 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.308339 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.316182 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.328013 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.335969 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.348013 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.356561 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.367951 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.376034 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.388070 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.395956 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.408445 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.415943 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.428041 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.436064 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.448135 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.455952 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.467980 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.475940 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.487996 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.495921 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.508149 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.516253 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.528281 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.535923 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.543988 IP 192.168.0.10.netbios-ns > 192.168.0.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
16:50:12.548478 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.555954 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.567954 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.575939 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.588117 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.595921 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.607974 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.615900 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.627981 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.635963 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.645900 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.647982 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
16:50:12.665951 IP 192.168.0.1.49204 > 192.168.0.10.49182: UDP, length 45
16:50:12.668234 IP 192.168.0.10.49182 > 192.168.0.1.49204: UDP, length 45
```

Figure 29. Test 3: Packet Capture on eth0 of NAT 1

E.4. NAT 1 Eth1

The following is a snapshot of the packets captured on the second interface (eth1) of NAT 1.

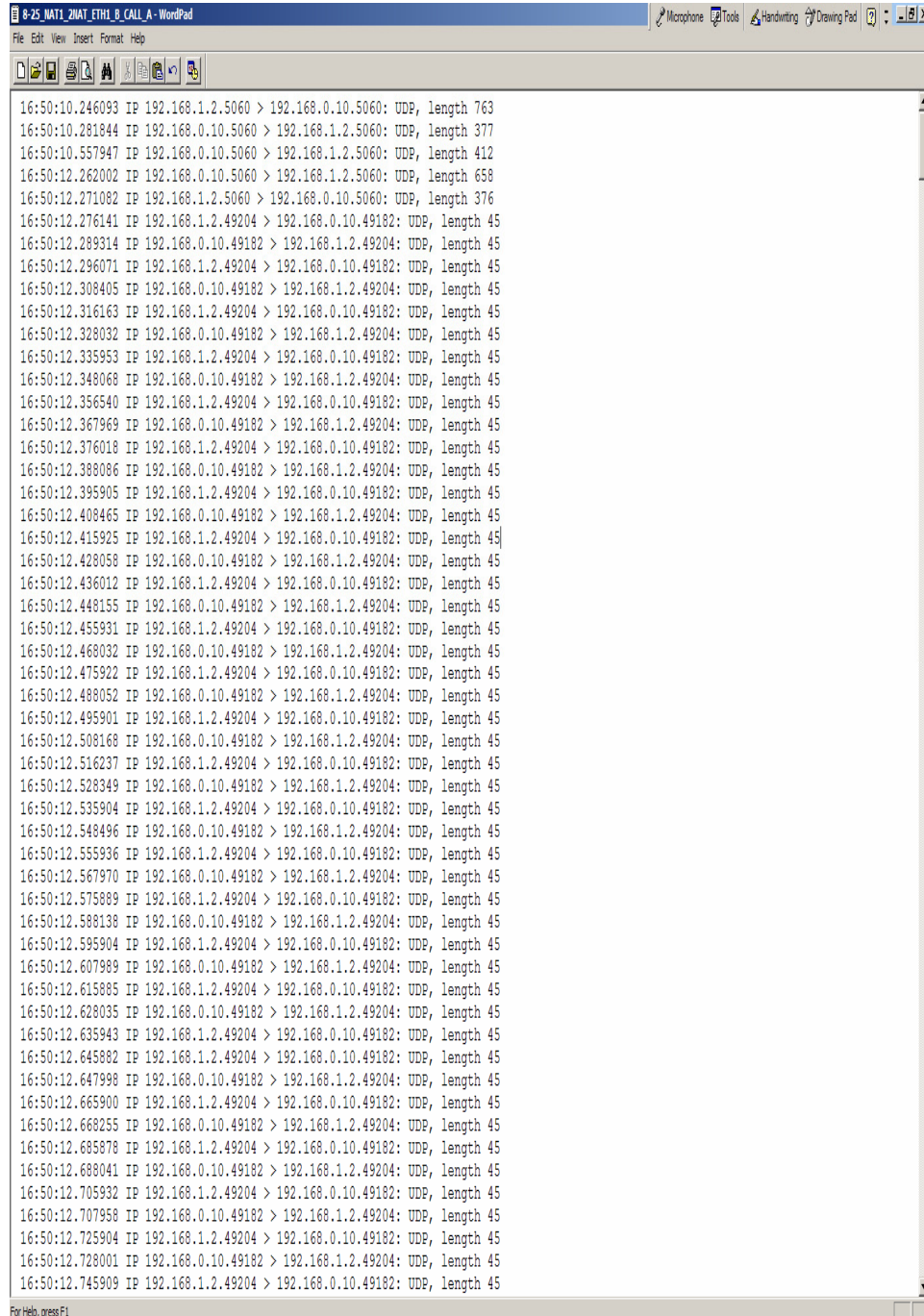


Figure 30. Test 3: Packet Capture on eth1 of NAT 1

E.5. NAT 2 Eth0

The following is a snapshot of the packets captured on the first interface (eth0) of NAT 2.

The screenshot shows a Wireshark capture of network traffic on interface eth0. The packet list pane displays 28 packets, starting with a SIP INVITE and followed by several RTP packets. The packet details pane for packet 1 (SIP) shows the full SIP message structure, including headers and body. The packet bytes pane at the bottom shows the raw hex and ASCII data of the selected packet.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.2	192.168.0.10	SIP/50	Request: INVITE sip:192.168.0.10, with session description
2	0.036816	192.168.0.10	192.168.1.2	SIP	Status: 100 Trying
3	0.312954	192.168.0.10	192.168.1.2	SIP	Status: 180 Ringing
4	0.017209	192.168.0.10	192.168.1.2	SIP/50	Status: 200 OK, with session description
5	2.025309	192.168.1.2	192.168.0.10	SIP	Request: ACK sip:192.168.0.10:5060
6	2.030636	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=316040833, Seq=12039, Time=0, Mark
7	2.044036	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=264990281, Seq=1034, Time=160
8	2.050565	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=316040833, Seq=12040, Time=160
9	2.063108	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=264990281, Seq=1035, Time=320
10	2.070656	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=316040833, Seq=12041, Time=320
11	2.082739	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=264990281, Seq=1036, Time=480
12	2.090447	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=316040833, Seq=12042, Time=480
13	2.102785	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=264990281, Seq=1037, Time=640
14	2.111032	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=316040833, Seq=12043, Time=640
15	2.122672	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=264990281, Seq=1038, Time=800
16	2.130511	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=316040833, Seq=12044, Time=800
17	2.142787	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=264990281, Seq=1039, Time=960
18	2.150296	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=316040833, Seq=12045, Time=960
19	2.163166	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=264990281, Seq=1040, Time=1120
20	2.170420	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=316040833, Seq=12046, Time=1120
21	2.182763	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=264990281, Seq=1041, Time=1280
22	2.190502	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=316040833, Seq=12047, Time=1280
23	2.202856	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=264990281, Seq=1042, Time=1440
24	2.210426	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=316040833, Seq=12048, Time=1440
25	2.222740	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=264990281, Seq=1043, Time=1600
26	2.230417	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=316040833, Seq=12049, Time=1600
27	2.242756	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=264990281, Seq=1044, Time=1760
28	2.250297	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=316040833, Seq=12050, Time=1760

Frame 1 (805 bytes on wire (805 bytes captured))
Ethernet II, Src: 192.168.0.1 (00:00:29:67:9e:9c), Dst: Linksys:ef:af:eb (00:0c:41:ef:af:eb)
Internet Protocol, Src: 192.168.1.2 (192.168.1.2), Dst: 192.168.0.10 (192.168.0.10)
User Datagram Protocol, Src Port: 5060 (5060), Dst Port: 5060 (5060)
Source port: 5060 (5060)
Destination port: 5060 (5060)
Length: 771
Checksum: 0x89ae [correct]
Session Initiation Protocol
Request-Line: INVITE sip:192.168.0.10 SIP/2.0
Message Header
Message body
Session Description Protocol
Session Description Protocol version (v): 0
Owner/Creator, Session Id (o): - 3334024859 3334024859 IN IP4 192.168.2.10
Session Name (s): S3phone
Connection Information (c): IN IP4 192.168.2.10
Time Description, active time (t): 0 0
Session Attribute (a): direction:active
Media Description, name and address (m): audio 49204 RTP/AVP 3 97 98 8 0 101
Media Attribute (a): rtptime:3 GSM/8000
Media Attribute (a): rtptime:97 fLBC/8000
Media Attribute (a): rtptime:98 fLBC/8000
Media Attribute (a): fmtp:98 mode=20
Media Attribute (a): rtptime:8 PCMA/8000

0000 00 0c 41 ef af eb 00 e0 29 67 9e 9c 08 00 45 00 ..A....)g....E.
0010 03 17 2b 5b 00 00 7f 11 8b 1e c0 a8 01 02 c0 a8 ..+{.....
0020 00 0a 13 c4 13 c4 03 03 89 ae 49 4e 56 49 54 45INVITE
0030 20 73 69 70 3a 31 39 32 2e 31 36 38 2e 30 2e 31 sip:192.168.0.1
0040 30 20 53 49 50 2f 32 2e 30 0d 0a 56 69 61 3a 20 0 SIP/2.0..Via:
0050 52 40 56 7f 27 2a 2a 2f 55 44 50 7a 21 20 27 2a SIP/2.0;branch=1

File: %J:\2 NAT\8-25 NAT 2(Eth0 2 NAT B Call) [P: 992; 992 M; 0

Figure 31. Test 3: Packet Capture on eth0 of NAT 2

E.6. NAT 2 Eth1

The following is a snapshot of the packets captured on the first interface (eth0) of NAT 2.

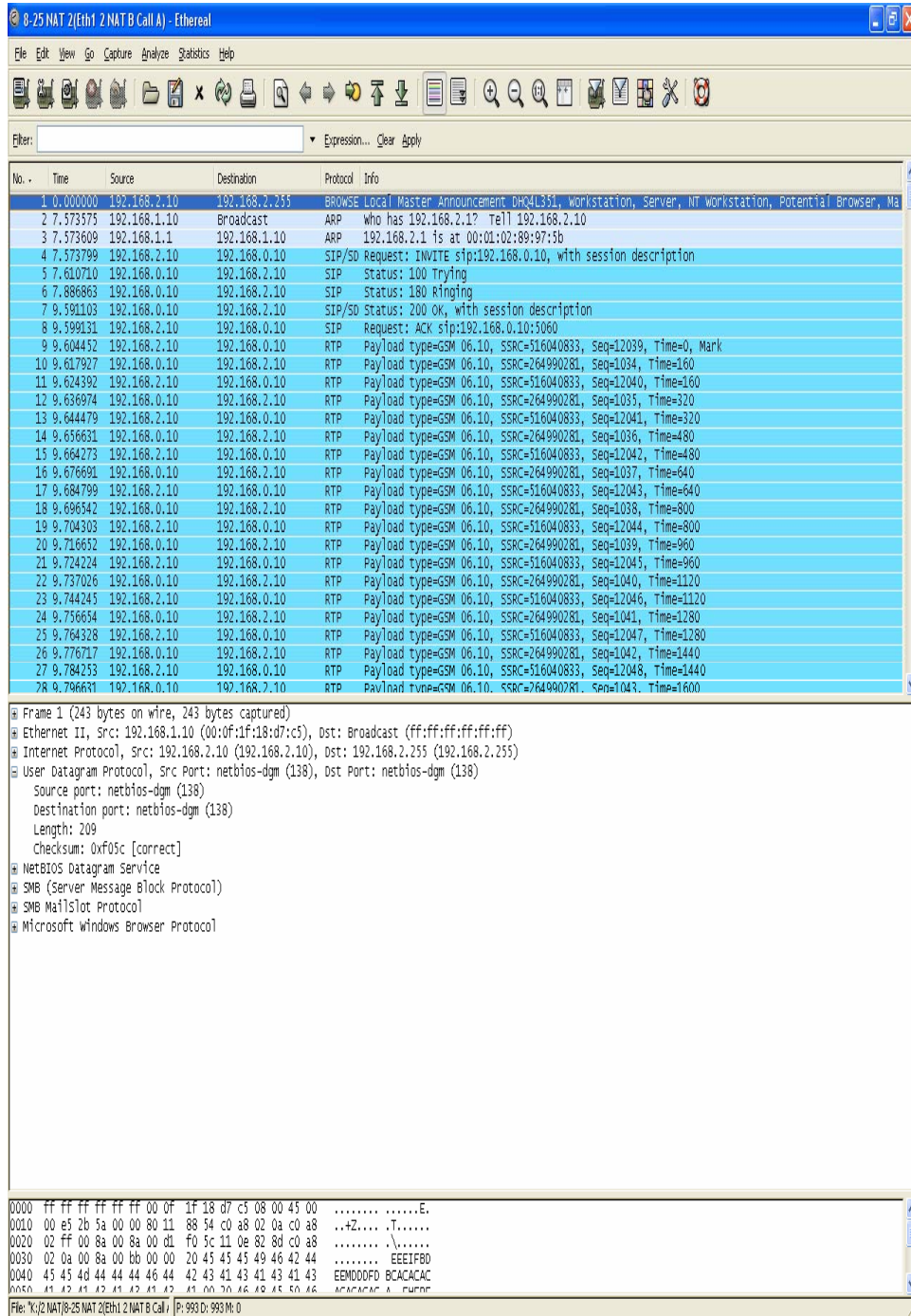


Figure 32. Test 3: Packet Capture on eth1 of NAT 2

E.7. Analysis

This demonstration was very similar to the one described in Appendix C. It differed in that now Client B is located behind two NAT devices instead of one. In other words, two layers of network address translation occurred before any packet can be moved between Client A and Client B.

The “INVITE” message (red outline in Figure 27) indicated that Client B will be sending and receiving RTP packets at 192.168.2.10 on port 49204 (purple outline in Figure 28). Client A acknowledged the invitation by sending a “200 OK” message to Client B with embedded SDP information indicating that it would send and receive RTP packets at 192.168.2.10 on port 49182 (green outline in Figure 27). Figures 27 and 29 show that Client A sends and receives RTP packet directly to/from the public IP address of NAT 1. As explained in Appendix C, SJPhone will first attempt to send RTP media packets to the IP address indicated in the SDP message (or the private IP address of Client B). Since the firewall installed on Client A was configured to drop packets destined to for Client B, none of the packets sent out by Client A ever reached Client B. Therefore, Client A resorted to sending subsequent RTP packets to the IP address from which it received Client B’s RTP media packets (blue outline in Figure 27). In this case, the packets were sent to the public IP address of the NAT 1 device (192.168.0.1). Figures 29 and 30 confirmed that the configured NAT operated correctly.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E. TEST 4: EXTENDED DOUBLE NAT VOIP DEMONSTRATION USING SJPHONE

The instructions contained in this appendix describe how to setup and demonstrate a SIP-based VoIP communication between two SIP-enabled clients via Network Address Translation (NAT) devices. In this setup, NAT 1 is no longer configured with a DNAT rule to rewrite the destination IP address of the packets that traverse it. Therefore, NAT 1 only performs SNAT. NAT 2 and NAT 3 are each configured with both SNAT and DNAT rules. The demonstration is conducted as follows: Client B initiates a call to Client A. Then Client C initiates a call to Client A after the VoIP session between Client B and A is terminated. Packet captures from all three clients and the NATs are included at the end of this appendix along with an analysis.

A. Network Topology

Refer to Figure 13 and Figure 14 for the physical and logical network topology.

B. Equipment Requirements

B.1. Clients A, B and C

B.1.1. Windows XP Operating System

B.1.2. Sound card

B.1.3. SJPhone v.1.60

B.1.4. Ethereal

B.1.5. ZoneAlarm (Client A only)

B.2. NAT 1, NAT 2 and NAT 3

B.2.1. Linux Operating System (Fedora Core 4)

B.2.2. netfilter and iptables

B.2.3. Ethereal

B.2.4. Two network cards

B.3. Additional Equipment

B.3.1. Cross-over cables and a switch or hub to implement the network architecture illustrated in Figure 13

B.3.2. Microphones as audio input devices for clients A, B, and C

C. Installation and Configuration

C.1. Client A

IP Address: 192.168.0.10

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.0.1

C.2. Client B

IP Address: 192.168.2.10

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.2.1

C.3. Client C

IP Address: 192.168.3.10

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.3.1

C.4. NAT 1

C.4.1. Configure eth0 by editing /etc/sysconfig/network-scripts/ifcfg-eth0:

DEVICE=eth0

BOOTPROTO=NONE

IPADDR=192.168.0.1

NETMASK=255.255.255.0

C.4.2. Activate eth0 by running:

ifup eth0

C.4.3. Configure eth1 by editing /etc/sysconfig/network-scripts/ifcfg-eth1:

DEVICE=eth1

BOOTPROTO=NONE

IPADDR=192.168.1.1

NETMASK=255.255.255.0

C.4.4. Activate eth1 by running:

ifup eth1

C.4.5. Enable IP Forwarding by running:

echo 1 > /proc/sys/net/ipv4/ip_forward

C.4.6. Flush any existing firewall and NAT rules by running:

```
iptables -F  
iptables -t nat -F
```

C.4.7. Configure NAT rule by running:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 192.168.0.1
```

C.5. NAT 2

C.5.1. Configure eth0 by editing /etc/sysconfig/network-scripts/ifcfg-eth0:

```
DEVICE=eth0  
BOOTPROTO=NONE  
IPADDR=192.168.1.2  
NETMASK=255.255.255.0  
GATEWAY=192.168.1.1
```

C.5.2. Activate eth0 by running:

```
ifup eth0
```

C.5.3. Configure eth1 by editing /etc/sysconfig/network-scripts/ifcfg-eth1:

```
DEVICE=eth1  
BOOTPROTO=NONE  
IPADDR=192.168.2.1  
NETMASK=255.255.255.0
```

C.5.4. Activate eth1 by running:

```
ifup eth1
```

C.5.5. Enable IP forwarding by running:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

C.5.6. Flush any existing firewall and NAT rules by running:

```
iptables -F  
iptables -t nat -F
```

C.5.7. Configure NAT rules by running:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 192.168.1.2  
iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 192.168.2.10
```

C.6. NAT 3

C.6.1. Configure eth0 by editing /etc/sysconfig/network-scripts/ifcfg-eth0:

```
DEVICE=eth0
```

```
BOOTPROTO=NONE
IPADDR=192.168.1.3
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
```

C.6.2. Activate eth0 by running:

```
ifup eth0
```

C.6.3. Configure eth1 by editing and saving `/etc/sysconfig/network-scripts/ifcfg-eth1`:

```
DEVICE=eth1
BOOTPROTO=NONE
IPADDR=192.168.3.1
NETMASK=255.255.255.0
```

C.6.4. Activate eth1 by running:

```
ifup eth1
```

C.6.5. Enable IP forwarding by running:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

C.6.6. Flush any existing firewall and NAT rules by running:

```
iptables -F
iptables -t nat -F
```

C.6.7. Configure NAT rules by running:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 192.168.1.3
iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 192.168.3.10
```

C.7. SJPhone Installation and Configuration

C.7.1. Clients A, B, and C

C.7.1.1. Download the Windows version of SJPhone v.1.60 from SJ Labs

C.7.1.2. Install SJPhone v.1.60

C.7.1.3. Launch SJPhone

C.7.1.3.1. On SJPhone,

C.7.1.3.1.1. Right-click

C.7.1.3.1.2. Go to **Services**

C.7.1.3.1.3. Select **PC-to-PC (SIP)**

C.8. Ethereal Installation and Configuration

C.8.1. Clients A, B, and C

C.8.1.1. Download the Windows version of Ethereal v.0.10.12

C.8.1.2. Install Ethereal v.0.10.12

C.8.2. NAT 2 and 3

C.8.2.1. Install Ethereal if it is not already installed

C.8.2.1.1. Go to the Desktop menu

C.8.2.1.2. Go to System Settings

C.8.2.1.3. Go to Add/Remove Applications

C.8.2.1.4. Click on Details under System Tools

C.8.2.1.5. Find and then check `ethereal-gnome`

C.8.2.1.6. Click on Close

C.8.2.1.7. Click on Update

C.8.2.1.8. Put in the correct Fedora Core 4 CDs when prompted

C.8.2.1.9.

C.9. ZoneAlarm Installation and Configuration

C.9.1. On client A,

C.9.1.1. Download the free ZoneAlarm from Zone Labs

C.9.1.2. Install ZoneAlarm by following on-screen instructions

C.9.1.3. When ZoneAlarm is being run for the first time, it will ask the user to choose between Basic ZoneAlarm or trial version of ZoneAlarm Pro, select the trial version of ZoneAlarm

C.9.1.4. Answer on-screen questions

C.9.1.5. When asked to select a security level for the detected network, select Allow into Trusted Zone

C.9.1.6. Configure firewall rule in ZoneAlarm:

C.9.1.6.1. Go to Firewall menu on the left panel

C.9.1.6.2. Click on the Expert tab

C.9.1.6.3. Click on Add

C.9.1.6.4. Type in a name for the firewall rule in the Name textbox

C.9.1.6.5. Under Action, select Block

C.9.1.6.6. Under **Destination**,

C.9.1.6.6.1. Select **Modify**

C.9.1.6.6.2. Select **Add Location**

C.9.1.6.6.3. Select **IP Address**

C.9.1.6.6.4. Type in a description in the **Description** textbox

C.9.1.6.6.5. Type 192.168.2.10 in the **IP Address** textbox

C.9.1.6.6.6. Click **OK**

C.9.1.6.7. Repeat steps C.9.1.6.1 to C.9.1.6.6.6 to create a rule to block
192.168.3.10

C.9.1.6.8. Click **OK**

C.9.1.6.9. Click **Apply**

D. Preparation and Testing

D.1. Adjust volume on both clients accordingly

D.2. Plug microphones into both clients

D.3. On Client A,

D.3.1. Launch **Ethereal**

D.3.2. Go to the **Capture** menu

D.3.3. Go to **Interfaces**

D.3.4. Click on **Capture 192.168.0.10**

D.4. On Client B,

D.4.1. Launch **Ethereal**

D.4.2. Go to the **Capture** menu

D.4.3. Go to **Interfaces**

D.4.4. Click on **Capture 192.168.2.10**

D.5. On NAT 1,

D.5.1. Launch one terminal and run:

`tcpdump -n -i eth0`

D.5.2. Launch another terminal and run:

`tcpdump -n -i eth1`

D.6. On NAT 2 and NAT 3,

D.6.1. Launch one instance of **Ethereal**

- D.6.1.1. Go to the **Capture** menu
 - D.6.1.2. Go to **Interfaces**
 - D.6.1.3. Click on **Capture Eth0**
- D.6.2. Launch another instance of **Ethereal**
 - D.6.2.1. Go to the **Capture** menu
 - D.6.2.2. Go to **Interfaces**
 - D.6.2.3. Click on **Capture Eth1**
- D.7. On Client B,
 - D.7.1. Call A by dialing 192.168.0.10 in SJPhone
- D.8. On Client A,
 - D.8.1. Select **Accept** in the pop-up dialog box when SJPhone rings
 - D.8.2. Clients A and B may engage in a VoIP conversation at this point.
 - D.8.3. Click on the **Hang-Up** bottom on either SJPhone to terminate call when finished
- D.9. On Client A,
 - Stop tcpdump packet captures by pressing **Control-C**
- D.10. On Client A, Client B, NAT 1, NAT 2 and NAT 3,
 - D.10.1. Stop packet captures by selecting **Stop** on **Ethereal**
 - D.10.2. Stop packet captures on NAT Box 1 by pressing **Control-C**
 - D.10.3. Repeat steps D.3 to D.8 for Clients A, Client C , NATs 1 and NAT 3

E. Packet Captures

E.1. Client B Calls A

E.1.1. Client A

The following is a snapshot of the packets captured on Client A when Client B calls Client A.

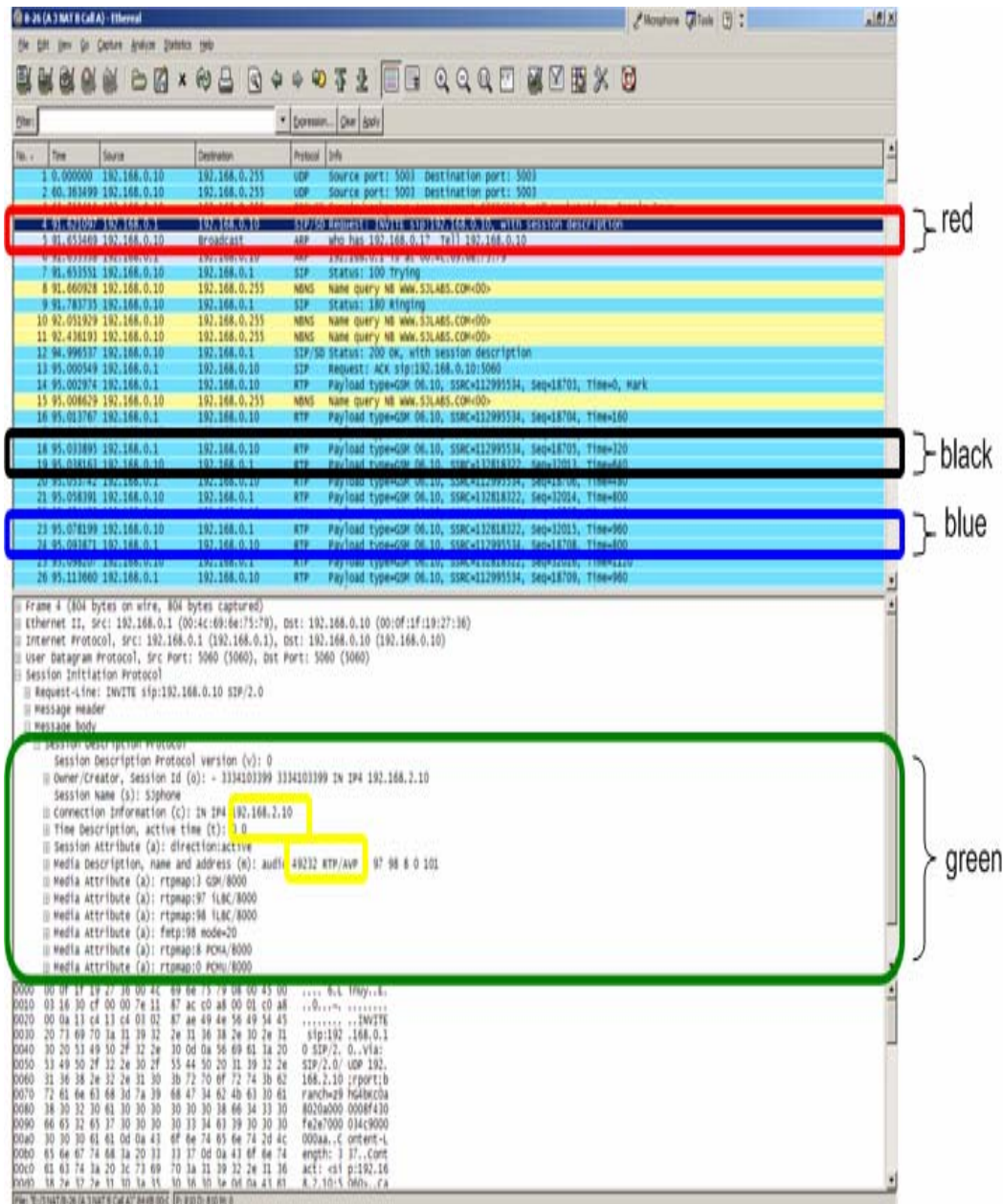


Figure 33. Test 4: Packet Capture on Client A (Client B Calls Client A)

E.1.2. Client B

The following is a snapshot of the packets captured on Client B when Client B calls Client A.

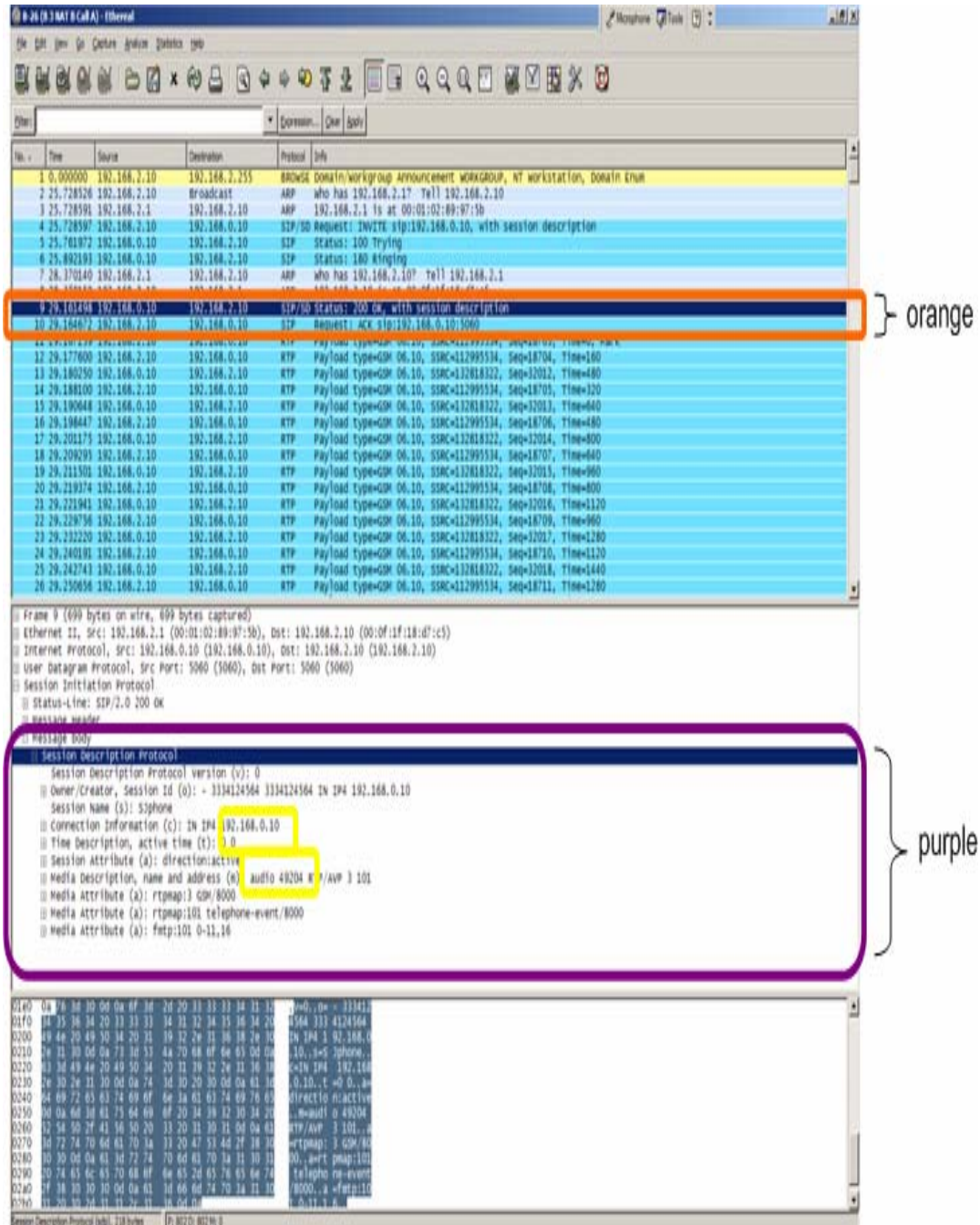
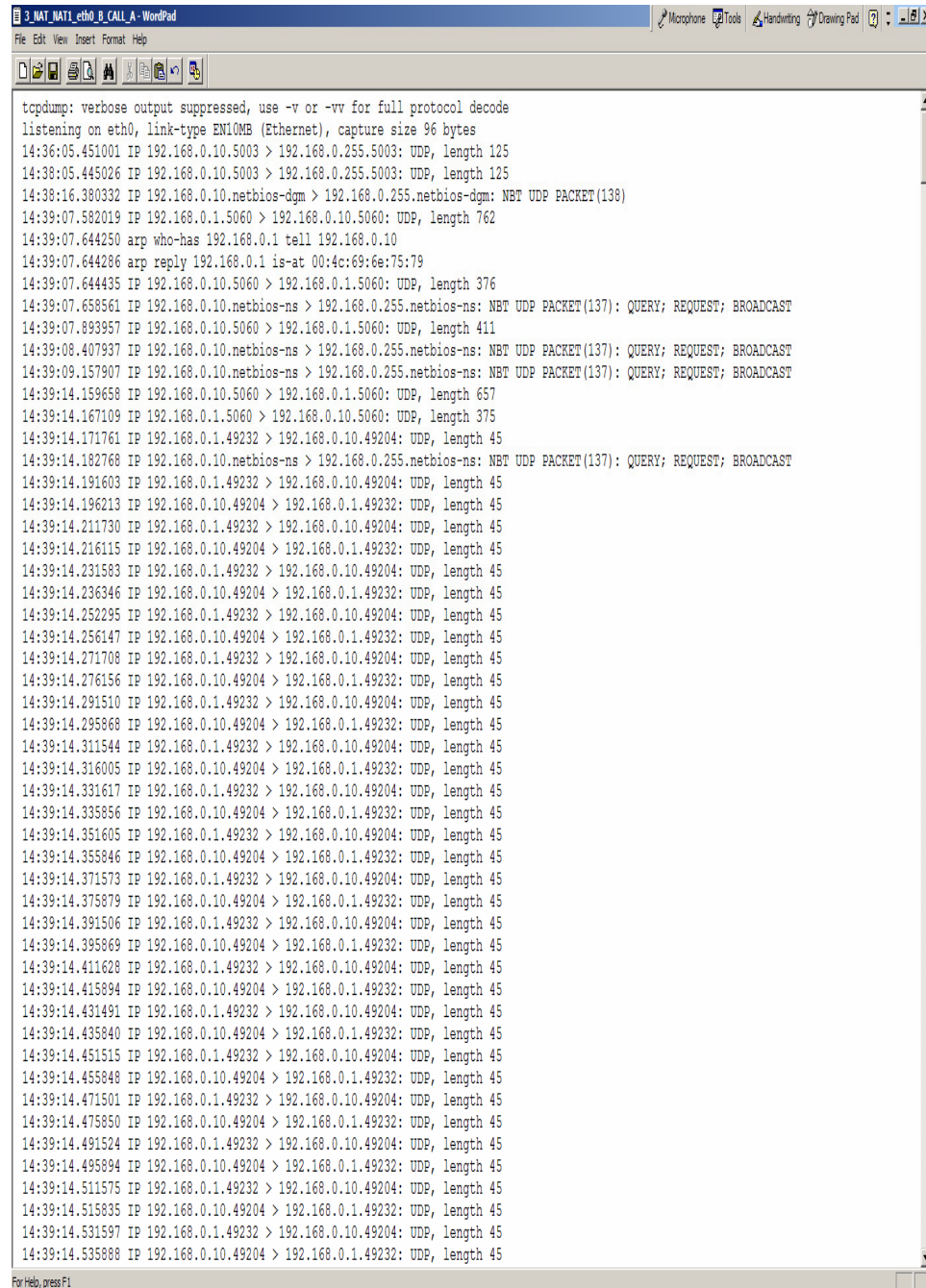


Figure 34. Test 4: Packet Capture on Client B (Client B Calls Client A)

E.1.3. NAT 1 Eth0

The following is a snapshot of the packets captured on the first interface (eth0) of NAT 1 when Client B calls Client A.

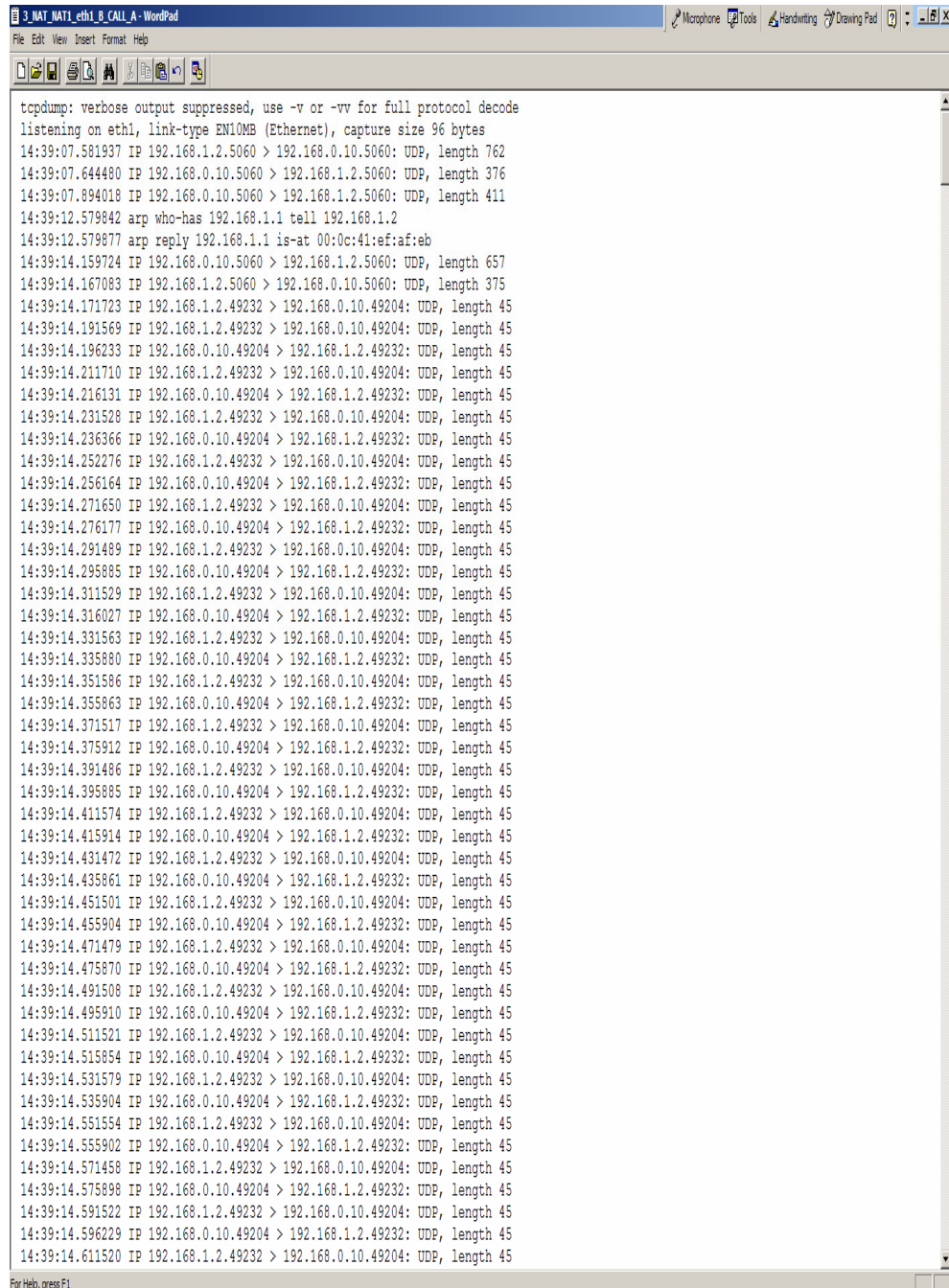


```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
14:36:05.451001 IP 192.168.0.10.5003 > 192.168.0.255.5003: UDP, length 125
14:38:05.445026 IP 192.168.0.10.5003 > 192.168.0.255.5003: UDP, length 125
14:38:16.380332 IP 192.168.0.10.netbios-dgm > 192.168.0.255.netbios-dgm: NBT UDP PACKET(138)
14:39:07.582019 IP 192.168.0.1.5060 > 192.168.0.10.5060: UDP, length 762
14:39:07.644250 arp who-has 192.168.0.1 tell 192.168.0.10
14:39:07.644286 arp reply 192.168.0.1 is-at 00:4c:69:6e:75:79
14:39:07.644435 IP 192.168.0.10.5060 > 192.168.0.1.5060: UDP, length 376
14:39:07.658561 IP 192.168.0.10.netbios-ns > 192.168.0.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
14:39:07.893957 IP 192.168.0.10.5060 > 192.168.0.1.5060: UDP, length 411
14:39:08.407937 IP 192.168.0.10.netbios-ns > 192.168.0.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
14:39:09.157907 IP 192.168.0.10.netbios-ns > 192.168.0.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
14:39:14.159658 IP 192.168.0.10.5060 > 192.168.0.1.5060: UDP, length 657
14:39:14.167109 IP 192.168.0.1.5060 > 192.168.0.10.5060: UDP, length 375
14:39:14.171761 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.182768 IP 192.168.0.10.netbios-ns > 192.168.0.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
14:39:14.191603 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.196213 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
14:39:14.211730 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.216115 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
14:39:14.231583 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.236346 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
14:39:14.252295 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.256147 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
14:39:14.271708 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.276156 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
14:39:14.291510 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.295868 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
14:39:14.311544 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.316005 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
14:39:14.331617 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.335856 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
14:39:14.351605 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.355846 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
14:39:14.371573 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.375879 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
14:39:14.391506 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.395869 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
14:39:14.411628 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.415894 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
14:39:14.431491 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.435840 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
14:39:14.451515 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.455848 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
14:39:14.471501 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.475850 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
14:39:14.491524 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.495894 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
14:39:14.511575 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.515835 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
14:39:14.531597 IP 192.168.0.1.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.535888 IP 192.168.0.10.49204 > 192.168.0.1.49232: UDP, length 45
```

Test 4: Packet Capture on eth0 of NAT 1 (Client B Calls Client A)

E.1.4. NAT 1 Eth1

The following is a snapshot of the packets captured on the second interface (eth1) of NAT 1 when Client B calls Client A.



```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
14:39:07.581937 IP 192.168.1.2.5060 > 192.168.0.10.5060: UDP, length 762
14:39:07.644480 IP 192.168.0.10.5060 > 192.168.1.2.5060: UDP, length 376
14:39:07.894018 IP 192.168.0.10.5060 > 192.168.1.2.5060: UDP, length 411
14:39:12.579842 arp who-has 192.168.1.1 tell 192.168.1.2
14:39:12.579877 arp reply 192.168.1.1 is-at 00:0c:41:ef:af:eb
14:39:14.159724 IP 192.168.0.10.5060 > 192.168.1.2.5060: UDP, length 657
14:39:14.167083 IP 192.168.1.2.5060 > 192.168.0.10.5060: UDP, length 375
14:39:14.171723 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.191569 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.196233 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.211710 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.216131 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.231528 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.236366 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.252276 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.256164 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.271650 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.276177 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.291489 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.295885 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.311529 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.316027 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.331563 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.335880 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.351586 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.355863 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.371517 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.375912 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.391486 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.395885 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.411574 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.415914 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.431472 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.435861 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.451501 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.455904 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.471479 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.475870 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.491508 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.495910 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.511521 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.515854 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.531579 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.535904 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.551554 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.555902 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.571458 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.575898 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.591522 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
14:39:14.596229 IP 192.168.0.10.49204 > 192.168.1.2.49232: UDP, length 45
14:39:14.611520 IP 192.168.1.2.49232 > 192.168.0.10.49204: UDP, length 45
```

Figure 35. Test 4: Packet Capture on eth1 of NAT 1 (Client B Calls Client A)

E.1.5. NAT 2 Eth0

The following is a snapshot of the packets captured on the first interface (eth0) of NAT 2 when Client B calls Client A.

The screenshot shows a Wireshark capture of network traffic on the interface 'eth0' of a host named 'NAT 2'. The capture filter is 'eth0'. The packet list shows 26 packets. The first packet is a SIP INVITE request from 192.168.1.2 to 192.168.0.10. The subsequent packets are SIP responses (100 Trying, 180 Ringing, 200 OK) and a series of RTP packets (GSM audio) from 192.168.1.2 to 192.168.0.10. The packet details pane shows the structure of the first packet (Frame 1), including Ethernet II, Internet Protocol, User Datagram Protocol, and Session Initiation Protocol (SIP) fields. The SIP message body is expanded, showing the Session Description Protocol (SDP) details, including session name, connection information, and media attributes for audio (GSM/8000).

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.2	192.168.0.10	SIP/SD	Request: INVITE sip:192.168.0.10, with session description
2	0.063612	192.168.0.10	192.168.1.2	SIP	Status: 100 Trying
3	0.313181	192.168.0.10	192.168.1.2	SIP	Status: 180 Ringing
4	4.998509	192.168.1.2	192.168.1.2	ARP	who has 192.168.1.1? Tell 192.168.1.2
5	4.998727	192.168.1.1	192.168.1.2	ARP	192.168.1.1 is at 00:0c:41:ef:af:eb
6	6.379104	192.168.0.10	192.168.1.2	SIP/SD	Status: 200 OK, with session description
7	6.385479	192.168.1.2	192.168.0.10	SIP	Request: ACK sip:192.168.0.10:5060
8	6.590388	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18703, Time=0, Mark
9	6.610229	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18704, Time=160
10	6.615116	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=132818322, Seq=32012, Time=480
11	6.630368	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18705, Time=320
12	6.635017	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=132818322, Seq=32013, Time=640
13	6.650186	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18706, Time=480
14	6.655252	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=132818322, Seq=32014, Time=800
15	6.670942	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18707, Time=640
16	6.675038	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=132818322, Seq=32015, Time=960
17	6.690304	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18708, Time=800
18	6.695052	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=132818322, Seq=32016, Time=1120
19	6.710255	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18709, Time=960
20	6.714759	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=132818322, Seq=32017, Time=1280
21	6.730192	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18710, Time=1120
22	6.734903	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=132818322, Seq=32018, Time=1440
23	6.750213	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18711, Time=1280
24	6.754757	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=132818322, Seq=32019, Time=1600
25	6.770251	192.168.1.2	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18712, Time=1440
26	6.774743	192.168.0.10	192.168.1.2	RTP	Payload type=GSM 06.10, SSRC=132818322, Seq=32020, Time=1760

Frame 1 (804 bytes on wire (804 bytes captured))
Ethernet II, Src: 192.168.1.2 (00:e0:29:67:9e:9c), Dst: 192.168.1.1 (00:0c:41:ef:af:eb)
Internet Protocol, Src: 192.168.1.2 (192.168.1.2), Dst: 192.168.0.10 (192.168.0.10)
User Datagram Protocol, Src Port: 5060 (5060), Dst Port: 5060 (5060)
Session Initiation Protocol
Request-Line: INVITE sip:192.168.0.10 SIP/2.0
Message Header
Message body
Session Description Protocol
Session Description Protocol Version (V): 0
Owner/Creator, Session Id (O): - 3334103399 3334103399 IN IP4 192.168.2.10
Session Name (S): S3phone
Connection Information (C): IN IP4 192.168.2.10
Time Description, active time (T): 0 0
Session Attribute (A): direction:active
Media Description, name and address (M): audio 49232 RTP/AVP 3 97 98 8 0 101
Media Attribute (A): rtpmap:3 GSM/8000
Media Attribute (A): rtpmap:97 iLBC/8000
Media Attribute (A): rtpmap:98 iLBC/8000
Media Attribute (A): fmtp:98 mode=20
Media Attribute (A): rtpmap:8 PCMA/8000
Media Attribute (A): rtpmap:0 PCMU/8000

0000 00 0c 41 ef af eb 00 e0 29 67 9e 9c 08 00 45 00 ..A....)g....E.
0010 03 16 30 cf 00 00 7f 11 85 ab c0 a8 01 02 c0 a8 ..0.....
0020 00 0a 13 c4 13 c4 03 02 86 ad 49 4e 56 49 54 45 INVITE
0030 20 73 69 70 3a 31 39 32 2e 31 36 38 2e 30 2e 31 sip:192.168.0.1
0040 30 20 53 49 50 2f 32 2e 30 0d 0a 56 69 61 3a 20 0 SIP/2.0..Via:
0050 53 49 50 2f 32 2e 30 2f 55 44 50 20 31 39 32 2e SIP/2.0/uDP 192.
0060 31 36 38 2e 32 2e 31 30 3b 72 70 6f 72 74 3b 62 168.2.10 ;rport:b
0070 72 61 6e 63 68 3d 7a 39 68 47 34 62 4b 63 30 61 ranch=z9 hG4bxc0a
0080 38 30 32 30 61 30 30 30 30 30 38 66 34 33 30 8020a000 0008f430
0090 66 65 32 65 37 30 30 30 30 33 34 63 39 30 30 30 fe2e7000 034c9000
00a0 30 30 30 61 61 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 00aa..c ontent-L
00b0 65 6e 67 74 68 3a 20 33 33 37 0d 0a 43 6f 6e 74 length: 3 37..Cont
00c0 61 63 74 3a 20 3c 73 69 70 3a 31 39 32 2e 31 36 act: <sl p:192.16
00d0 38 7e 37 7e 31 30 3a 35 30 36 30 3e 0d 0a 43 61 8.2.10:5 060>..ca

Figure 36. Test 4: Packet Capture on eth0 of NAT 2 (Client B Calls Client A)

E.1.6. NAT 2 Eth1

The following is a snapshot of the packets captured on the second interface (eth1) of NAT 2 when Client B calls Client A.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.2.10	192.168.2.255	BROWSE	Domain/workgroup Announcement WORKGROUP, NT workstation, Domain Enum
2	49.310980	192.168.2.10	Broadcast	ARP	who has 192.168.2.1? Tell 192.168.2.10
3	49.311019	192.168.2.1	192.168.2.10	ARP	192.168.2.1 is at 00:01:02:89:97:5b
4	49.311212	192.168.2.10	192.168.0.10	SIP/SD	Request: INVITE sip:192.168.0.10, with session description
5	49.374948	192.168.0.10	192.168.2.10	SIP	Status: 100 Trying
6	49.624525	192.168.0.10	192.168.2.10	SIP	Status: 180 Ringing
7	54.373778	192.168.2.1	192.168.2.10	ARP	who has 192.168.2.10? Tell 192.168.2.1
8	54.373890	192.168.2.10	192.168.2.1	ARP	192.168.2.10 is at 00:0f:1f:18:d7:c5
9	55.890437	192.168.0.10	192.168.2.10	SIP/SD	Status: 200 OK, with session description
10	55.896727	192.168.2.10	192.168.0.10	SIP	Request: ACK sip:192.168.0.10:5060
11	55.901637	192.168.2.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18703, Time=0, Mark
12	55.921455	192.168.2.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18704, Time=160
13	55.926444	192.168.0.10	192.168.2.10	RTP	Payload type=GSM 06.10, SSRC=132818322, Seq=32012, Time=480
14	55.941584	192.168.2.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18705, Time=320
15	55.946368	192.168.0.10	192.168.2.10	RTP	Payload type=GSM 06.10, SSRC=132818322, Seq=32013, Time=640
16	55.961408	192.168.2.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18706, Time=480
17	55.966551	192.168.0.10	192.168.2.10	RTP	Payload type=GSM 06.10, SSRC=132818322, Seq=32014, Time=800
18	55.982195	192.168.2.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18707, Time=640
19	55.986343	192.168.0.10	192.168.2.10	RTP	Payload type=GSM 06.10, SSRC=132818322, Seq=32015, Time=960
20	56.001529	192.168.2.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18708, Time=800
21	56.006352	192.168.0.10	192.168.2.10	RTP	Payload type=GSM 06.10, SSRC=132818322, Seq=32016, Time=1120
22	56.021413	192.168.2.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18709, Time=960
23	56.026054	192.168.0.10	192.168.2.10	RTP	Payload type=GSM 06.10, SSRC=132818322, Seq=32017, Time=1280
24	56.041416	192.168.2.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18710, Time=1120
25	56.046205	192.168.0.10	192.168.2.10	RTP	Payload type=GSM 06.10, SSRC=132818322, Seq=32018, Time=1440
26	56.061471	192.168.2.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=112995534, Seq=18711, Time=1280

Frame 1 (251 bytes on wire (251 bytes captured))

- Ethernet II, Src: 192.168.2.10 (00:0f:1f:18:d7:c5), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- Internet Protocol, Src: 192.168.2.10 (192.168.2.10), Dst: 192.168.2.255 (192.168.2.255)
- User Datagram Protocol, Src Port: netbios-dgm (138), Dst Port: netbios-dgm (138)
- NetBIOS Datagram Service
- SMB (Server Message Block Protocol)
- SMB Mailslot Protocol
- Microsoft Windows Browser Protocol

0000 ff ff ff ff ff ff ff 1f 18 d7 c5 08 00 45 00E.
0010 00 ed 30 ce 00 00 80 11 82 08 c0 a8 02 0a c0 a8 ..0.....
0020 02 ff 00 8a 00 8a 00 d9 5b a0 11 0e 83 40 c0 a8[...&..
0030 02 0a 00 8a 00 c3 00 00 20 45 45 45 49 46 42 44EEEIFBD
0040 45 45 4d 44 44 44 44 46 44 42 43 41 43 41 43 41 43 EEMDDDFD BCACACAC
0050 41 43 41 43 41 43 41 41 00 20 41 42 41 43 46 ACACACAA A. ABACF
0060 50 46 50 45 46 46 44 45 43 46 43 45 50 46 48 46 PFPFNDFE CFCEPFHF
0070 44 45 46 46 50 46 50 41 43 41 42 00 ff 53 4d 42 DEFFFPFA CAB..SMB
0080 25 00 00 00 00 00 00 00 00 00 00 00 00 00 00 %.....
0090 00 00 00 00 00 00 00 00 00 00 00 11 00 00 29
00a0 00 00 00 00 00 00 00 00 e8 03 00 00 00 00 00
00b0 00 00 29 00 56 00 03 00 01 00 01 02 00 3a ...).V.....
00c0 00 5c 4d 41 49 4c 53 4c 4f 54 5c 42 52 4f 57 53 .\MAILSL OT\BROWS
00d0 45 00 0c 00 a0 hh 0d 00 57 4f 52 4b 47 52 4f 55 F.....WORKGROU

File: E:\3\NAT\8-26 (Eth1.3\NAT 8 Call A)*83\KB 0 [P:803 D:803 M:0

Figure 37. Test 4: Packet Capture on eth1 of NAT 2 (Client B Calls Client A)

E.1.7. Analysis

As soon as Client B dialed the IP address of Client A, Client B sent out an “INVITE” message to Client A (red outline in Figure 33). The message had embedded SDP information to inform Client A that Client B would be sending and receiving RTP packets at 192.168.2.10 on port 49232 (purple outline in Figure 34). To acknowledge the invitation, Client A sent a “200 OK” packet to Client B with embedded SDP information indicating that it would send and receive RTP packets at 192.168.2.10 on port 49204 (green outline in Figure 33). The packets captured on Client A indicate that Client A sent and received RTP packet directly to/from the public IP address of NAT 1. As explained in Appendix C, SJPhone will first attempt to send RTP media packets to the IP address indicated in the SDP message (or the private IP address of Client B). Since the firewall installed on Client A was configured to drop packets destined to the private IP address of Client B, none of the packets sent out by Client A could reach Client B. Therefore, Client A then sent subsequent RTP packets to the IP address in which received the RTP media packets (blue outline in Figure 34)

The packet captures indicate that the first RTP packet is sent by Client B (black outline in Figure 33). Even though NAT 1 was not explicitly configured to rewrite the destination IP address of incoming packets to 192.168.1.2 (public IP address of NAT 2), NAT 1 intelligently does this on its own. A reasonable explanation for this behavior is that *iptables* in NAT 1 maintained information for packets that are initiated from the local network [19]. In our scenario, the first RTP packet is processed according to the SNAT rule when it arrives at NAT 1. At the same time, NAT 1 created an entry in its connection tracking table to store essential information (such as source and destination IP addresses and ports) that would allow it to associate incoming packets with the session between Client A and Client B. Packets determined to belong to a certain packet previously received from NAT 2 (due to SNAT) had their destination IP address changed to the public IP address of NAT 2 (refer to Figures 35 through 38). This is evident from observing the packet captures on eth1 of NAT 2.

E.2. Client C Calls Client A

E.2.1. Client A

The following is a snapshot of the packets captured on Client A when Client C calls Client A.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.10	192.168.0.255	UDP	Source port: 5003 Destination port: 5003
2	60.176582	192.168.0.10	192.168.0.255	UDP	Source port: 5003 Destination port: 5003
3	62.423265	192.168.0.1	Broadcast	ARP	Who has 192.168.0.10? Tell 192.168.0.1
4	62.423278	192.168.0.1	192.168.0.1	ARP	192.168.0.10 is at 00:0f:1f:19:27:36
5	62.423351	192.168.0.1	192.168.0.10	SIP/SDP	Request: INVITE sip:192.168.0.10, with session description
6	62.457945	192.168.0.10	192.168.0.1	SIP	Status: 100 Trying
7	62.469516	192.168.0.10	192.168.0.255	NBNS	Name query NB www.SJLABS.COM<00>
8	62.590691	192.168.0.10	192.168.0.1	SIP	Status: 180 Ringing
9	62.858943	192.168.0.10	192.168.0.255	NBNS	Name query NB www.SJLABS.COM<00>
10	63.245316	192.168.0.10	192.168.0.255	NBNS	Name query NB www.SJLABS.COM<00>
11	65.081662	192.168.0.10	192.168.0.1	SIP/SDP	Status: 200 OK, with session description
12	65.084592	192.168.0.1	192.168.0.10	SIP	Request: ACK sip:192.168.0.10:5060
13	65.089033	192.168.0.10	192.168.0.255	NBNS	Name query NB www.SJLABS.COM<00>
14	65.093222	192.168.0.1	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18868, Time=0, Mark
15	65.095716	192.168.0.10	192.168.0.1	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10101, Time=160
16	65.103673	192.168.0.1	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18869, Time=160
17	65.105557	192.168.0.10	192.168.0.1	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10102, Time=320
18	65.114026	192.168.0.1	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18870, Time=320
19	65.115980	192.168.0.10	192.168.0.1	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10103, Time=480
20	65.124500	192.168.0.1	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18871, Time=480
21	65.126414	192.168.0.10	192.168.0.1	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10104, Time=640
22	65.134890	192.168.0.1	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18872, Time=640
23	65.137048	192.168.0.10	192.168.0.1	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10105, Time=800
24	65.145403	192.168.0.1	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18873, Time=800
25	65.147296	192.168.0.10	192.168.0.1	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10106, Time=960
26	65.15763	192.168.0.1	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18874, Time=960

Frame 5 (803 bytes on wire, 803 bytes captured)

Ethernet II, Src: 192.168.0.1 (00:4c:69:6e:75:79), Dst: 192.168.0.10 (00:0f:1f:19:27:36)

Internet Protocol, Src: 192.168.0.1 (192.168.0.1), Dst: 192.168.0.10 (192.168.0.10)

User Datagram Protocol, Src Port: 5060 (5060), Dst Port: 5060 (5060)

Session Initiation Protocol

Request-Line: INVITE sip:192.168.0.10 SIP/2.0

Message Header

Message body

Session Description Protocol

Session Description Protocol Version (v): 0

Owner/Creator, Session Id (o): - 3334083070 3334083070 IN IP4 192.168.3.10

Session Name (s): sjphone

Connection Information (c): IN IP4 192.168.3.10

Time Description, active time (t): 0 0

Session Attribute (a): direction:active

Media Description, name and address (m): audio 49156 RTP/AVP 3 97 98 8 0 101

Media Attribute (a): rtpmap:3 GSM/8000

Media Attribute (a): rtpmap:97 iLBC/8000

Media Attribute (a): rtpmap:98 iLBC/8000

Media Attribute (a): fmtp:98 mode=20

Media Attribute (a): rtpmap:8 PCMA/8000

Media Attribute (a): rtpmap:0 PCMU/8000

0000 00 0f 1f 19 27 36 00 4c 69 6e 75 79 08 00 45 006.L inuy..E.

0010 03 15 06 22 00 00 7e 11 b2 5a c0 a8 00 01 c0 a8Z.....

0020 00 0a 13 c4 13 c4 03 01 63 1e 49 4e 56 49 54 45c.INVITE

0030 20 73 69 70 3a 31 39 32 2e 31 36 38 2e 30 2e 31 sip:192.168.0.1

0040 30 20 53 49 50 2f 32 2e 30 0d 0a 56 69 61 3a 20 0 SIP/2.0.Via:

0050 53 49 50 2f 32 2e 30 2f 55 44 50 20 31 39 32 2e SIP/2.0/UDP 192.

0060 31 36 38 2e 33 2e 31 30 36 72 70 6f 72 74 3b 62 168.3.10;rport:b

0070 72 61 6e 63 68 3d 7a 39 68 47 34 62 4b 63 30 61 ranch=29 hg4bkcd0a

0080 38 30 33 30 61 30 30 30 30 30 31 64 34 33 30 8030a000 0001d430

0090 66 39 33 37 65 30 30 30 30 34 37 66 65 30 30 30 f937e000 047fe000

00a0 30 30 30 39 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 00009..c content-L

00b0 65 6e 67 74 68 3a 20 33 33 37 0d 0a 43 6f 6e 74 ength: 3 37..Cont

00c0 61 63 74 3a 20 3c 73 69 70 3a 31 39 32 2e 31 36 act: <sl p:192.16

00d0 38 2e 33 2e 31 30 3a 35 30 36 30 3e 0d 0a 43 61 8.3.10:5 0600..ca

Figure 38. Test 4: Packet Capture on Client A (Client C Calls Client A)

E.2.2. Client C

The following is a snapshot of the packets captured on Client C when Client C calls Client A.

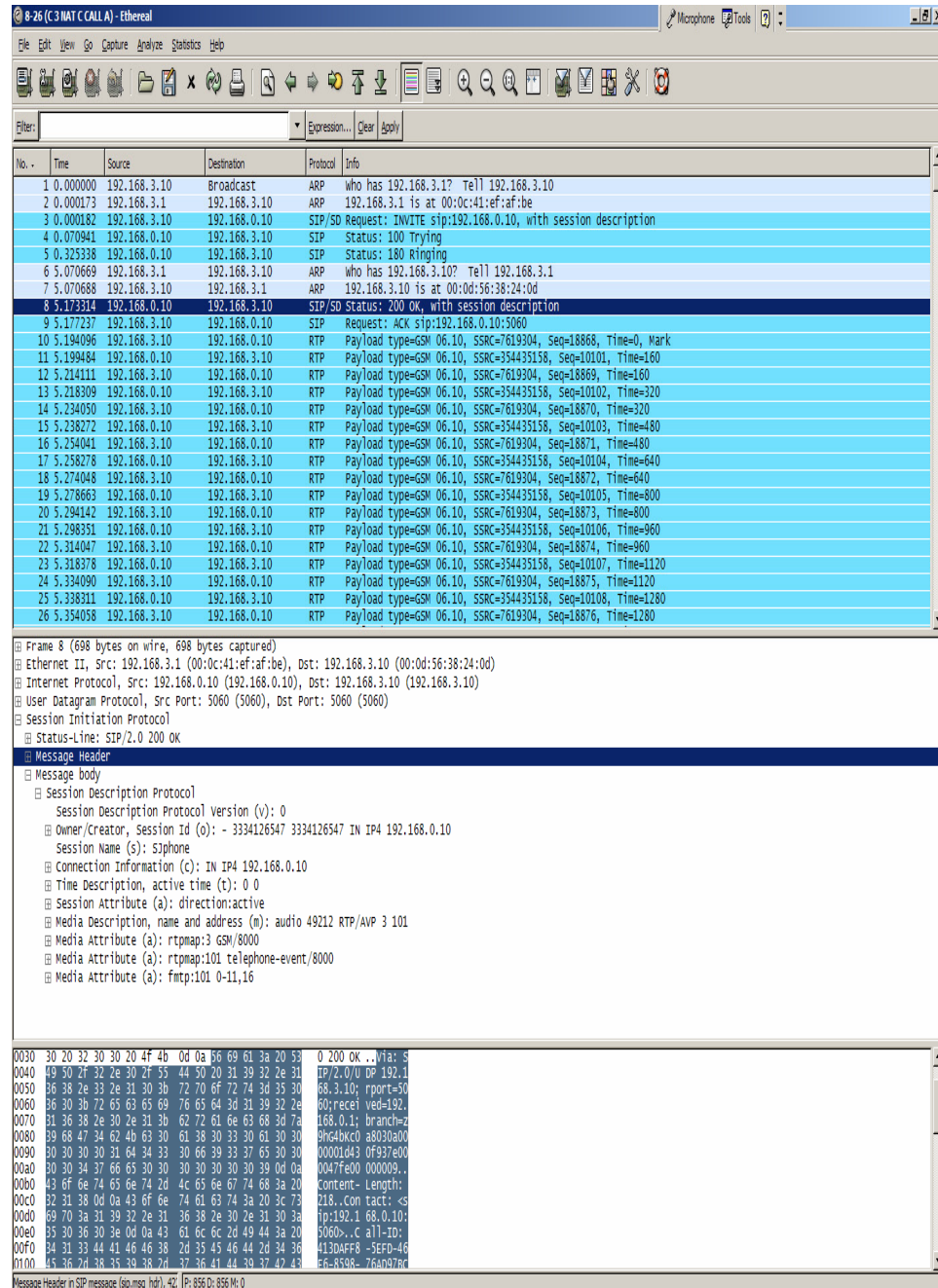
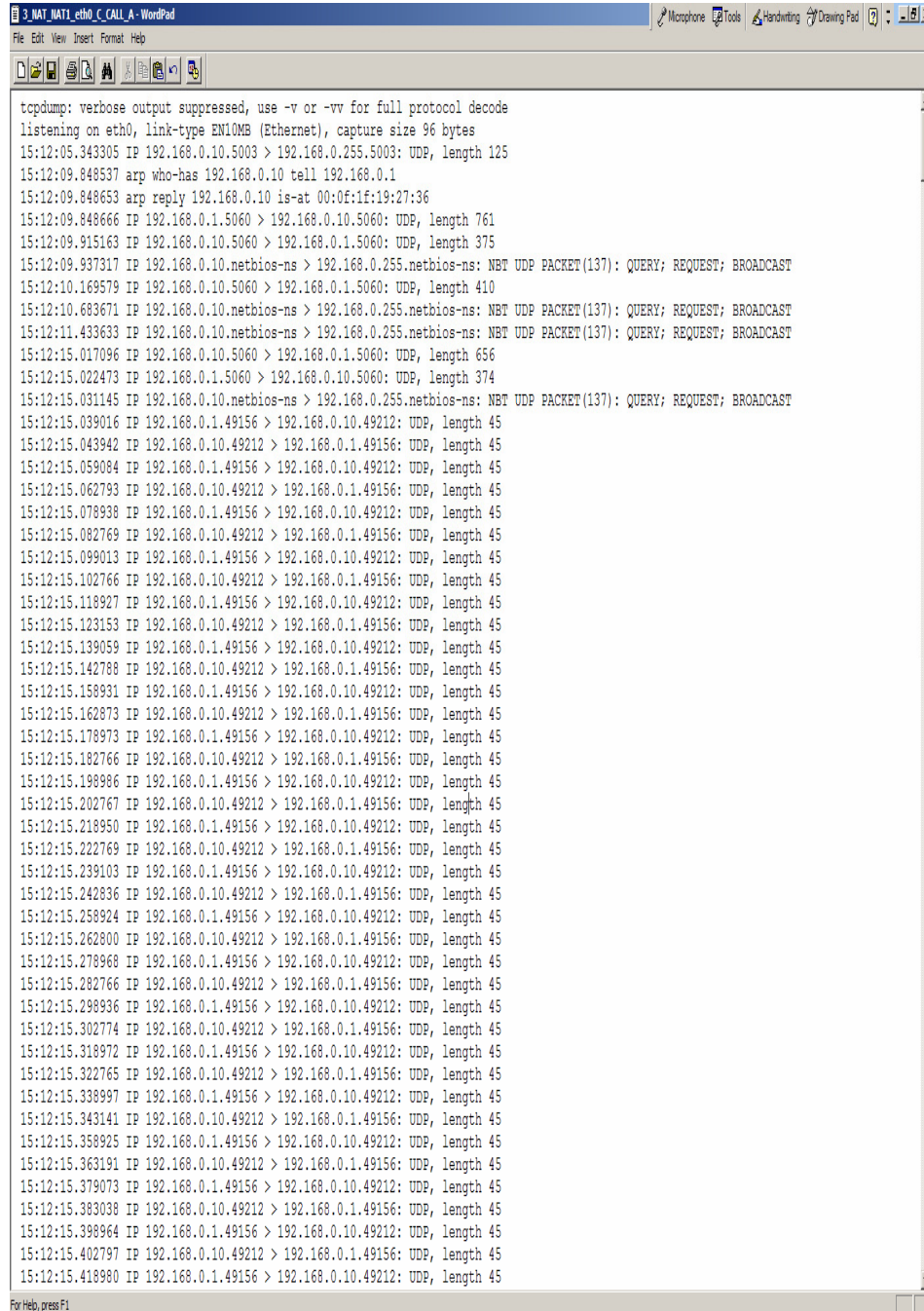


Figure 39. Test 4: Packet Capture on Client C (Client C Calls Client A)

E.2.3. NAT 1 Eth0

The following is a snapshot of the packets captured on the first interface (eth0) of NAT 1 when Client C calls Client A.

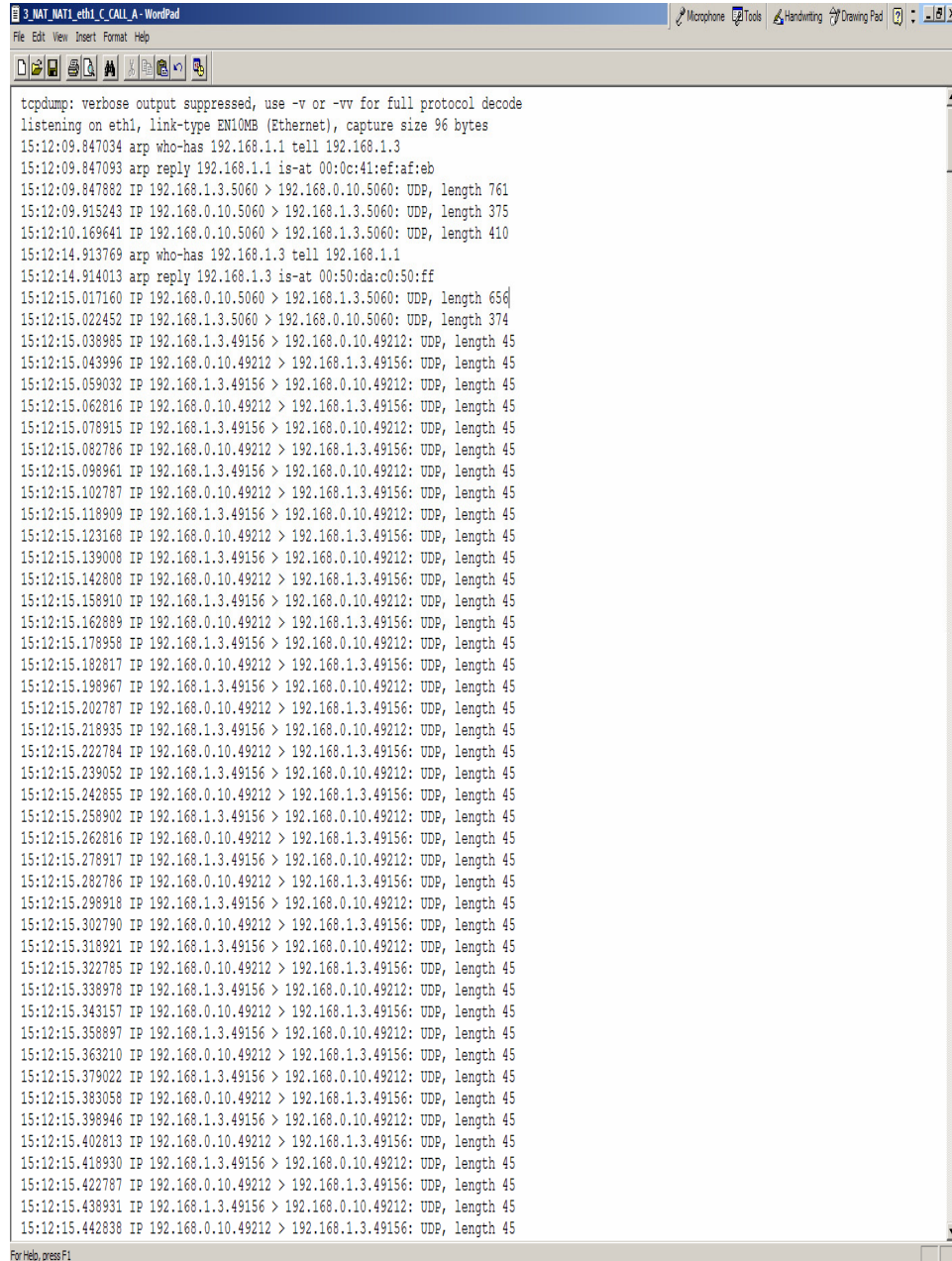


```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
15:12:05.343305 IP 192.168.0.10.5003 > 192.168.0.255.5003: UDP, length 125
15:12:09.848537 arp who-has 192.168.0.10 tell 192.168.0.1
15:12:09.848653 arp reply 192.168.0.10 is-at 00:0f:1f:19:27:36
15:12:09.848666 IP 192.168.0.1.5060 > 192.168.0.10.5060: UDP, length 761
15:12:09.915163 IP 192.168.0.10.5060 > 192.168.0.1.5060: UDP, length 375
15:12:09.937317 IP 192.168.0.10.netbios-ns > 192.168.0.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
15:12:10.169579 IP 192.168.0.10.5060 > 192.168.0.1.5060: UDP, length 410
15:12:10.683671 IP 192.168.0.10.netbios-ns > 192.168.0.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
15:12:11.433633 IP 192.168.0.10.netbios-ns > 192.168.0.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
15:12:15.017096 IP 192.168.0.10.5060 > 192.168.0.1.5060: UDP, length 656
15:12:15.022473 IP 192.168.0.1.5060 > 192.168.0.10.5060: UDP, length 374
15:12:15.031145 IP 192.168.0.10.netbios-ns > 192.168.0.255.netbios-ns: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
15:12:15.039016 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.043942 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.059084 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.062793 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.078938 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.082769 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.099013 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.102766 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.118927 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.123153 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.139059 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.142788 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.158931 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.162873 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.178973 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.182766 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.198986 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.202767 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.218950 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.222769 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.239103 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.242836 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.258924 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.262800 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.278968 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.282766 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.298936 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.302774 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.318972 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.322765 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.338997 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.343141 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.358925 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.363191 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.379073 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.383038 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.398964 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.402797 IP 192.168.0.10.49212 > 192.168.0.1.49156: UDP, length 45
15:12:15.418980 IP 192.168.0.1.49156 > 192.168.0.10.49212: UDP, length 45
```

Figure 40. Test 4: Packet Capture on eth0 of NAT 1 (Client C Calls Client A)

E.2.4. NAT 1 Eth1

The following is a snapshot of the packets captured on the second interface (eth1) of NAT 1 when Client C calls Client A.



```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
15:12:09.847034 arp who-has 192.168.1.1 tell 192.168.1.3
15:12:09.847093 arp reply 192.168.1.1 is-at 00:0c:41:ef:af:eb
15:12:09.847882 IP 192.168.1.3.5060 > 192.168.0.10.5060: UDP, length 761
15:12:09.915243 IP 192.168.0.10.5060 > 192.168.1.3.5060: UDP, length 375
15:12:10.169641 IP 192.168.0.10.5060 > 192.168.1.3.5060: UDP, length 410
15:12:14.913769 arp who-has 192.168.1.3 tell 192.168.1.1
15:12:14.914013 arp reply 192.168.1.3 is-at 00:50:da:c0:50:ff
15:12:15.017160 IP 192.168.0.10.5060 > 192.168.1.3.5060: UDP, length 656
15:12:15.022452 IP 192.168.1.3.5060 > 192.168.0.10.5060: UDP, length 374
15:12:15.038985 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.043996 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.059032 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.062816 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.078915 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.082786 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.098961 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.102787 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.118909 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.123168 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.139008 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.142808 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.158910 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.162889 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.178958 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.182817 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.198967 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.202787 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.218935 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.222784 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.239052 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.242855 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.258902 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.262816 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.278917 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.282786 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.298918 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.302790 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.318921 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.322785 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.338978 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.343157 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.358897 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.363210 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.379022 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.383058 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.398946 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.402813 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.418930 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.422787 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
15:12:15.438931 IP 192.168.1.3.49156 > 192.168.0.10.49212: UDP, length 45
15:12:15.442838 IP 192.168.0.10.49212 > 192.168.1.3.49156: UDP, length 45
```

Figure 41. Test 4: Packet Capture on eth1 of NAT 1 (Client C Calls Client A)

E.2.5. NAT 3 Eth0

The following is a snapshot of the packets captured on the first interface (eth0) of NAT 3 when Client C calls Client A.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.3	Broadcast	ARP	who has 192.168.1.1? Tell 192.168.1.3
2	0.000238	192.168.1.1	192.168.1.3	ARP	192.168.1.1 is at 00:0c:41:ef:af:eb
3	0.000256	192.168.1.3	192.168.0.10	SIP/SD	Request: INVITE sip:192.168.0.10, with session description
4	0.068677	192.168.0.10	192.168.1.3	SIP	Status: 100 Trying
5	0.323084	192.168.0.10	192.168.1.3	SIP	Status: 180 Ringing
6	5.066542	192.168.1.1	192.168.1.3	ARP	who has 192.168.1.3? Tell 192.168.1.1
7	5.066604	192.168.1.3	192.168.1.1	ARP	192.168.1.3 is at 00:50:da:c0:50:ff
8	5.170436	192.168.0.10	192.168.1.3	SIP/SD	Status: 200 OK, with session description
9	5.174745	192.168.1.3	192.168.0.10	SIP	Request: ACK sip:192.168.0.10:5060
10	5.191547	192.168.1.3	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18868, Time=0, Mark
11	5.196770	192.168.0.10	192.168.1.3	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10101, Time=160
12	5.211581	192.168.1.3	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18869, Time=160
13	5.215588	192.168.0.10	192.168.1.3	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10102, Time=320
14	5.231472	192.168.1.3	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18870, Time=320
15	5.235554	192.168.0.10	192.168.1.3	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10103, Time=480
16	5.251507	192.168.1.3	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18871, Time=480
17	5.255555	192.168.0.10	192.168.1.3	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10104, Time=640
18	5.271464	192.168.1.3	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18872, Time=640
19	5.275934	192.168.0.10	192.168.1.3	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10105, Time=800
20	5.291557	192.168.1.3	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18873, Time=800
21	5.295578	192.168.0.10	192.168.1.3	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10106, Time=960
22	5.311460	192.168.1.3	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18874, Time=960
23	5.315651	192.168.0.10	192.168.1.3	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10107, Time=1120
24	5.331508	192.168.1.3	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18875, Time=1120
25	5.335582	192.168.0.10	192.168.1.3	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10108, Time=1280
26	5.351511	192.168.1.3	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18876, Time=1280

Frame 1 (42 bytes on wire, 42 bytes captured)
 Ethernet II, Src: 192.168.1.3 (00:50:da:c0:50:ff), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Address Resolution Protocol (request)

```

0000  ff ff ff ff ff ff ff ff 00 50 da c0 50 ff 08 06 00 01 .....P..P....
0010  08 00 06 04 00 01 00 50 da c0 50 ff c0 a8 01 03 .....P..P....
0020  00 00 00 00 00 00 c0 a8 01 01 .....
  
```

File: E:\3\NAT\8-26 (3 NATS NAT 3 eth0 C Call A) | P: 893 D: 893 M: 0

Figure 42. Test 4: Packet Capture on eth0 of NAT 3 (Client C Calls Client A)

E.2.6. NAT 3 Eth1

The following is a snapshot of the packets captured on the second interface (eth1) of NAT 3 when Client C calls Client A.

The screenshot shows a Wireshark capture of network traffic on interface eth1 of NAT 3. The capture filter is set to 'eth1 C Call A'. The packet list shows 26 packets. The first packet is an ARP request from 192.168.3.1 to the broadcast address. The next two packets are SIP messages: an INVITE and a 100 Trying response. This is followed by a 180 Ringing response. Then, another ARP request is seen from 192.168.3.1. This is followed by a 200 OK SIP response and a SIP ACK. The remainder of the capture consists of RTP packets of type GSM, with sequence numbers ranging from 18868 to 18876 and timestamps from 0 to 1280. The packet details for the first packet (Frame 1) are expanded, showing it is an Ethernet II frame from 192.168.3.1 to the broadcast address, containing an ARP request. The packet bytes are displayed in hexadecimal and ASCII at the bottom.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.3.10	Broadcast	ARP	who has 192.168.3.1? Tell 192.168.3.10
2	0.000202	192.168.3.1	192.168.3.10	ARP	192.168.3.1 is at 00:0c:41:ef:af:be
3	0.000103	192.168.3.10	192.168.0.10	SIP/SD	Request: INVITE sip:192.168.0.10, with session description
4	0.070499	192.168.0.10	192.168.3.10	SIP	Status: 100 Trying
5	0.324894	192.168.0.10	192.168.3.10	SIP	Status: 180 Ringing
6	5.069694	192.168.3.1	192.168.3.10	ARP	who has 192.168.3.10? Tell 192.168.3.1
7	5.069825	192.168.3.10	192.168.3.1	ARP	192.168.3.10 is at 00:0d:56:38:24:0d
8	5.172230	192.168.0.10	192.168.3.10	SIP/SD	Status: 200 OK, with session description
9	5.176440	192.168.3.10	192.168.0.10	SIP	Request: ACK sip:192.168.0.10:5060
10	5.193226	192.168.3.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18868, Time=0, Mark
11	5.198511	192.168.0.10	192.168.3.10	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10101, Time=160
12	5.213241	192.168.3.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18869, Time=160
13	5.217334	192.168.0.10	192.168.3.10	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10102, Time=320
14	5.233169	192.168.3.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18870, Time=320
15	5.237296	192.168.0.10	192.168.3.10	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10103, Time=480
16	5.253164	192.168.3.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18871, Time=480
17	5.257302	192.168.0.10	192.168.3.10	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10104, Time=640
18	5.273161	192.168.3.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18872, Time=640
19	5.277676	192.168.0.10	192.168.3.10	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10105, Time=800
20	5.293260	192.168.3.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18873, Time=800
21	5.297364	192.168.0.10	192.168.3.10	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10106, Time=960
22	5.313156	192.168.3.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18874, Time=960
23	5.317392	192.168.0.10	192.168.3.10	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10107, Time=1120
24	5.333204	192.168.3.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18875, Time=1120
25	5.337323	192.168.0.10	192.168.3.10	RTP	Payload type=GSM 06.10, SSRC=354435158, Seq=10108, Time=1280
26	5.353169	192.168.3.10	192.168.0.10	RTP	Payload type=GSM 06.10, SSRC=7619304, Seq=18876, Time=1280

Frame 1 (60 bytes on wire (60 bytes captured))
 Ethernet II, Src: 192.168.3.10 (00:0d:56:38:24:0d), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Address Resolution Protocol (request)

```

0000 ff ff ff ff ff ff 00 0d 56 38 24 0d 08 06 00 01 ..... V8S....
0010 08 00 06 04 00 01 00 0d 56 38 24 0d c0 a8 03 0a ..... V8S....
0020 00 00 00 00 00 00 c0 a8 03 01 11 0e 80 66 c0 a8 .....f...
0030 03 0a 00 8a 00 bb 00 00 20 45 44 46 ..... EDF
  
```

File: E:\3\NAT\8-26 (3 NATS NAT 3 eth1 C Call A) P: 861 D: 861 M: 0

Figure 43. Test 4: Packet Capture on eth1 of NAT 3 (Client C Calls Client A)

E.2.7. Analysis

The sequences of packet exchanges between Clients A and C are similar to that of Clients A and B described in the previous subsection. See Section E.1.7 for explanation.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX F. TEST 5: EXTENDED DOUBLE NAT VOIP WITH SIMULTANEOUS VOIP SESSIONS DEMONSTRATION USING SJPHONE

The instructions contained in this appendix describe how to setup and demonstrate a SIP-based VoIP communication using the network topology illustrated in the Figure 13. In this test scenario, two VoIP communication sessions will take place simultaneously, i.e. one between Clients A and C and the other between Clients B and D. Similar to the previous test, the DNAT rule is purposely taken out from NAT 1. Packet captures from all four clients are included at the end of this appendix along with their corresponding analysis.

A. Network Topology

Refer to Figure 13 and Figure 14 for the physical and logical network topology.

B. Equipment Requirements

B.1. Clients A, B, C, and D

B.1.1. Windows XP Operating System

B.1.2. Sound card

B.1.3. SJPhone v.1.60

B.1.4. Ethereal

B.1.5. ZoneAlarm (for Clients A and B only)

B.2. NAT 1, NAT 2, NAT 3 and Router

B.2.1. Linux Operating System (Fedora Core 4)

B.2.2. netfilter and iptables

B.2.3. Ethereal

B.2.4. Two network cards (for NAT 1, NAT 2, and NAT 3)

B.2.5. Three network cards (for Router only)

B.3. Additional Equipment

B.3.1. Cross-over cables and a switch or hub to implement the network architecture illustrated in Figure 13.

B.3.2. Microphones as audio input devices for clients A, B, C and D

C. Installation and Configuration

C.1. Client A

IP Address: 131.120.9.16

Subnet Mask: 255.255.255.0

Default Gateway: 131.120.9.15

C.2. Client B

IP Address: 131.120.9.17

Subnet Mask: 255.255.255.0

Default Gateway: 131.120.9.15

C.3. Clients C and D

IP Address: 192.168.3.11

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.3.1

C.4. Router

C.4.1. Configure eth0 by editing /etc/sysconfig/network-scripts/ifcfg-eth0 to include the following:

DEVICE=eth0

BOOTPROTO=NONE

IPADDR=192.168.100.27

NETMASK=255.255.255.0

GATEWAY=192.168.100.88

C.4.2. Activate eth0 by running:

ifup eth0

C.4.3. Configure eth1 by editing /etc/sysconfig/network-scripts/ifcfg-eth1 to include the following:

DEVICE=eth1

BOOTPROTO=NONE

IPADDR=192.168.202.1

NETMASK=255.255.255.0

C.4.4. Activate eth1 by running:

ifup eth1

C.4.5. Configure eth2 by editing and saving `/etc/sysconfig/network-scripts/ifcfg-eth2` to include the following:

```
DEVICE=eth2
BOOTPROTO=NONE
IPADDR=192.168.2.1
NETMASK=255.255.255.0
```

C.4.6. Activate eth2 by running:

```
ifup eth2
```

C.4.7. Enable IP Forwarding by running:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

C.4.8. Flush any existing firewall and NAT rules by running:

```
iptables -F
iptables -t nat -F
```

C.5. NAT 1

C.5.1. Configure eth0 by editing `/etc/sysconfig/network-scripts/ifcfg-eth0` to include the following:

```
DEVICE=eth0
BOOTPROTO=NONE
IPADDR=131.120.9.15
NETMASK=255.255.255.0
```

C.5.2. Activate eth0 by running:

```
ifup eth0
```

C.5.3. Configure eth1 by editing and saving `/etc/sysconfig/network-scripts/ifcfg-eth1` to include the following:

```
DEVICE=eth1
BOOTPROTO=NONE
IPADDR=192.168.100.88
NETMASK=255.255.255.0
```

C.5.4. Activate eth1 by running:

```
ifup eth1
```

C.5.5. Configure static routes by running:

```
route add -net 192.168.202.0 netmask 255.255.255.0 gw 192.168.100.27 eth1
route add -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.100.27 eth1
```

C.5.6. Enable IP Forwarding by running:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

C.5.7. Flush any existing firewall and NAT rules by running:

```
iptables -F
iptables -t nat -F
```

C.5.8. Configure NAT rule by running:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 131.120.9.15
```

C.6. NAT 2

C.6.1. Configure eth0 by editing and saving /etc/sysconfig/network-scripts/ifcfg-eth0 to include the following:

```
DEVICE=eth0
BOOTPROTO=NONE
IPADDR=196.168.202.11
NETMASK=255.255.255.0
GATEWAY=198.168.202.1
```

C.6.2. Activate eth0 by running:

```
ifup eth0
```

C.6.3. Configure eth1 by editing and saving /etc/sysconfig/network-scripts/ifcfg-eth1 to include the following:

```
DEVICE=eth1
BOOTPROTO=NONE
IPADDR=192.168.3.1
NETMASK=255.255.255.0
```

C.6.4. Activate eth1 by running:

```
ifup eth1
```

C.6.5. Enable IP Forwarding by running:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

C.6.6. Flush any existing firewall and NAT rules by running:

```
iptables -F
iptables -t nat -F
```

C.6.7. Configure NAT rules by running:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 192.168.202.11
```

```
iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 192.168.3.11
```

C.7. NAT 3

C.7.1. Configure eth0 by editing `/etc/sysconfig/network-scripts/ifcfg-eth0` to include the following:

```
DEVICE=eth0
```

```
BOOTPROTO=NONE
```

```
IPADDR=192.168.2.11
```

```
NETMASK=255.255.255.0
```

```
GATEWAY=192.168.2.1
```

C.7.2. Activate eth0 by running:

```
ifup eth0
```

C.7.3. Configure eth1 by editing `/etc/sysconfig/network-scripts/ifcfg-eth1` to include the following:

```
DEVICE=eth1
```

```
BOOTPROTO=NONE
```

```
IPADDR=192.168.3.1
```

```
NETMASK=255.255.255.0
```

C.7.4. Activate eth1 by running:

```
ifup eth1
```

C.7.5. Enable IP Forwarding by running:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

C.7.6. Flush any existing firewall and NAT rules by running:

```
iptables -F
```

```
iptables -t nat -F
```

C.7.7. Configure NAT rules by running:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 192.168.2.11
```

```
iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 192.168.3.11
```

C.8. SJPhone Installation and Configuration

C.8.1. Clients A, B, C and D

C.8.1.1. Download the Windows version of SJPhone v.1.60 from SJ Labs

- C.8.1.2. Install SJPhone v.1.60
- C.8.1.3. Launch SJPhone
- C.8.1.4. Right-click on SJPhone
- C.8.1.5. Go to **Services**
- C.8.1.6. Select **PC-to-PC (SIP)**

C.9. Etherreal Installation and Configuration

C.9.1. Clients A, B, C and D

- C.9.1.1. Download the Windows version of Etherreal v.0.10.12
- C.9.1.2. Install Etherreal v.0.10.12 by following on-screen instructions

C.9.2. Router, NAT 2 and NAT 3

C.9.2.1.1. Install Etherreal if it is not already installed

- C.9.2.1.1.1. Go to the **Desktop** menu
- C.9.2.1.1.2. Go to **System Settings**
- C.9.2.1.1.3. Go to **Add/Remove Applications**
- C.9.2.1.1.4. Click on **Details** under **System Tools**
- C.9.2.1.1.5. Find and then check **etherreal-gnome**
- C.9.2.1.1.6. Click on **Close**
- C.9.2.1.1.7. Click on **Update**
- C.9.2.1.1.8. Put in the correct Fedora Core 4 CDs when prompted

C.10. ZoneAlarm Installation and Configuration

C.10.1. On clients A and B,

- C.10.1.1. Download the free ZoneAlarm from Zone Labs
- C.10.1.2. Install ZoneAlarm by following on-screen instructions
- C.10.1.3. When ZoneAlarm is being run for the first time, it will ask the user to choose between Basic ZoneAlarm or trial version of ZoneAlarm Pro, select the trial version of ZoneAlarm
- C.10.1.4. Answer on-screen questions
- C.10.1.5. Click to use **Trial Version**
- C.10.1.6. When asked to select a security level for the detected network, select **Allow into Trusted Zone**

C.10.1.7. Allow all pop-ups for testing

C.10.1.8. Configure firewall rule in ZoneAlarm:

C.10.1.8.1. Go to **Firewall** menu on the left panel

C.10.1.8.2. Click on the **Expert** tab

C.10.1.8.3. Click on **Add**

C.10.1.8.4. Type in a name for the firewall rule in the **Name** textbox

C.10.1.8.5. Under **Action**, select **Block**

C.10.1.8.6. Under **Destination**,

C.10.1.8.6.1. Select **Modify**

C.10.1.8.6.2. Select **Add Location**

C.10.1.8.6.3. Select **IP Address**

C.10.1.8.6.4. Type in a description in the **Description** textbox

C.10.1.8.6.5. Type 192.168.3.11 in the **IP Address** textbox

C.10.1.8.6.6. Click **OK**

C.10.1.8.6.7. Click **OK**

C.10.1.8.6.8. Click **Apply**

D. Preparation and Testing

D.1. Adjust volume on both clients accordingly

D.2. Plug microphones into all clients

D.3. On Client A,

D.3.1. Launch **Ethereal**

D.3.2. Go to the **Capture** menu

D.3.3. Go to **Interfaces**

D.3.4. Click on **Capture 131.120.9.16**

D.4. On Client B,

D.4.1. Launch **Ethereal**

D.4.2. Go to the **Capture** menu

D.4.3. Go to **Interfaces**

D.4.4. Click on **Capture 131.120.9.17**

D.5. On Client C and Client D,

D.5.1. Launch **Ethereal**

- D.5.2. Go to the **Capture** menu
- D.5.3. Go to **Interfaces**
- D.5.4. Click on **Capture 192.168.3.11**
- D.6. On Router,
 - D.6.1. Launch one instance of Ethereal
 - D.6.1.1. Go to the **Capture** menu
 - D.6.1.2. Go to **Interfaces**
 - D.6.1.3. Click on **Capture Eth0**
 - D.6.2. Launch a second instance of Ethereal
 - D.6.2.1. Go to the **Capture** menu
 - D.6.2.2. Go to **Interfaces**
 - D.6.2.3. Click on **Capture Eth1**
 - D.6.3. Launch a third instance of Ethereal
 - D.6.3.1. Go to the **Capture** menu
 - D.6.3.2. Go to **Interfaces**
 - D.6.3.3. Click on **Capture Eth2**
- D.7. On NAT 1, NAT 2 and NAT 3,
 - D.7.1. Launch one instance of Ethereal
 - D.7.1.1. Go to the **Capture** menu
 - D.7.1.2. Go to **Interfaces**
 - D.7.1.3. Click on **Capture Eth0**
 - D.7.2. Launch another instance of Ethereal
 - D.7.2.1. Go to the **Capture** menu
 - D.7.2.2. Go to **Interfaces**
 - D.7.2.3. Click on **Capture Eth1**
- D.8. On Client C,
 - D.8.1. Call A by dialing 131.120.9.16 in SJPhone
- D.9. On Client A,
 - D.9.1. Select **Accept** in the pop-up dialog box when SJPhone rings
 - D.9.2. Clients A and C may engage in a VoIP conversation at this point.
- D.10. On Client D,

- D.10.1. Call B by dialing 131.120.9.17 in SJPhone
- D.11. On Client B,
 - D.11.1. Select **Accept** in the pop-up dialog box when SJPhone rings
- D.12. Clients B and D may engage in a VoIP conversation at this point.
- D.13. Click on the Hang-Up button on either SJPhone to terminate call when finished
- D.14. On all clients, NATs and Router,
 - D.14.1. Stop packet captures selecting **Stop** on Ethereal

E. Packet Captures

E.1. Client A

The following is a snapshot of the packets captured on Client A when it receives a call from Client C.

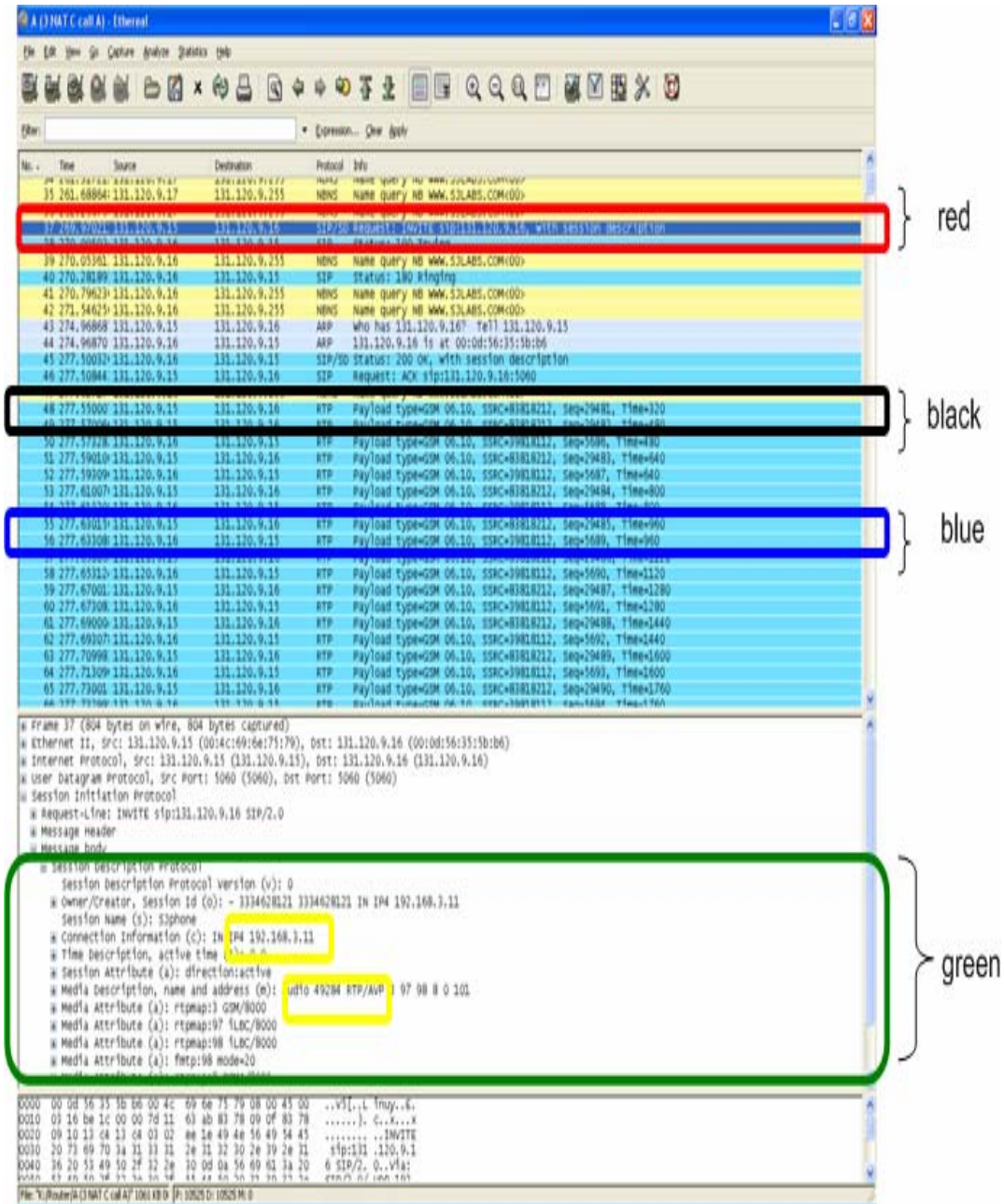


Figure 44. Test 5: Packet Capture on Client A

E.2. Client C

The following is a snapshot of the packets captured on Client C when it calls Client A.

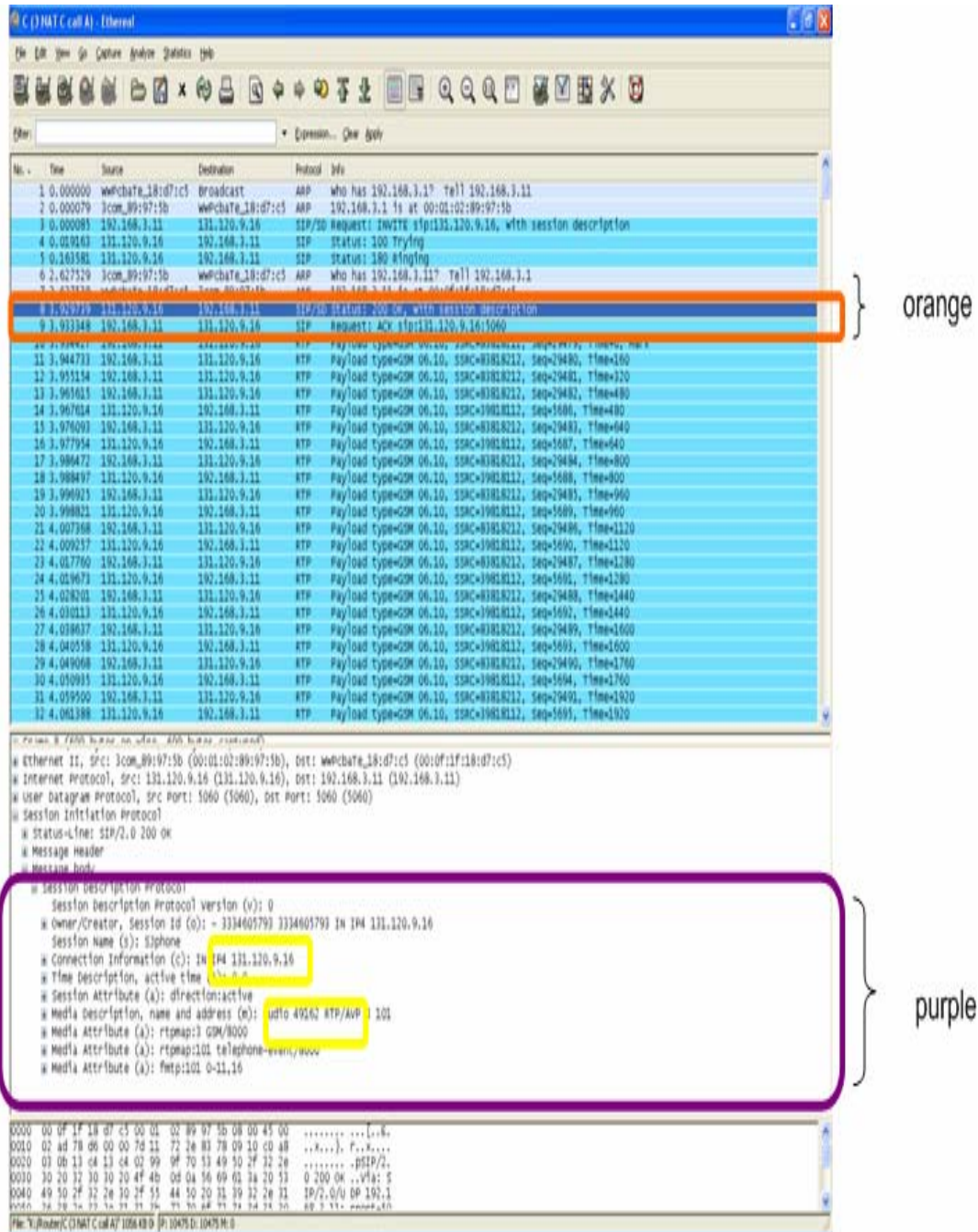


Figure 45. Test 5: Packet Capture on Client C

E.3. NAT 2 Eth0

The following is a snapshot of the packets captured on the first interface (eth0) of NAT 2.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.202.11	Broadcast	ARP	who has 192.168.202.1? Tell 192.168.202.11
2	0.000124	192.168.202.11	192.168.202.11	ARP	192.168.202.1 is at 00:01:02:89:97:6f
3	0.000135	192.168.202.11	131.120.9.16	SIP/SD	Request: INVITE sip:131.120.9.16, with session description
4	0.035911	131.120.9.16	192.168.202.11	SIP	Status: 100 Trying
5	0.312751	131.120.9.16	192.168.202.11	SIP	Status: 180 Ringing
6	5.034209	192.168.202.1	192.168.202.11	ARP	who has 192.168.202.11? Tell 192.168.202.1
7	5.034264	192.168.202.11	192.168.202.1	ARP	192.168.202.11 is at 00:e0:29:67:9e:9c
8	7.530899	131.120.9.16	192.168.202.11	SIP/SD	Status: 200 OK, with session description
9	7.538201	192.168.202.11	131.120.9.16	SIP	Request: ACK sip:131.120.9.16:5060
10	7.540185	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29479, Time=0, Mark
11	7.559939	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29480, Time=160
12	7.579923	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29481, Time=320
13	7.599964	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29482, Time=480
14	7.603583	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5686, Time=480
15	7.620042	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29483, Time=640
16	7.623390	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5687, Time=640
17	7.639941	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29484, Time=800
18	7.643602	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5688, Time=800
19	7.659966	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29485, Time=960
20	7.663401	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5689, Time=960
21	7.679990	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29486, Time=1120
22	7.683400	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5690, Time=1120
23	7.699905	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29487, Time=1280
24	7.703361	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5691, Time=1280
25	7.719918	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29488, Time=1440
26	7.723366	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5692, Time=1440
27	7.739920	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29489, Time=1600
28	7.743387	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5693, Time=1600
29	7.759904	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29490, Time=1760
30	7.763281	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5694, Time=1760
31	7.779903	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29491, Time=1920
32	7.783314	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5695, Time=1920

Frame 1 (42 bytes on wire (42 bytes captured))
Ethernet II, Src: 192.168.202.11 (00:e0:29:67:9e:9c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
 Hardware type: Ethernet (0x0001)
 Protocol type: IP (0x0800)
 Hardware size: 6
 Protocol size: 4
 Opcode: request (0x0001)
 Sender MAC address: 192.168.202.11 (00:e0:29:67:9e:9c)
 Sender IP address: 192.168.202.11 (192.168.202.11)
 Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
 Target IP address: 192.168.202.1 (192.168.202.1)

0000 ff ff ff ff ff ff 00 e0 29 67 9e 9c 08 06 00 01)g.....
0010 08 00 06 04 00 01 00 e0 29 67 9e 9c c0 a8 ca 0b)g.....
0020 00 00 00 00 00 00 c0 a8 ca 01

File: "C:\Router\NAT 2 Eth0 (3 NAT)" 1058 KB P: 10507 D: 10507 M: 0

Figure 46. Test 5: Packet Capture on eth0 of NAT 2

E.4. NAT 2 Eth1

The following is a snapshot of the packets captured on the second interface (eth1) of NAT 2.

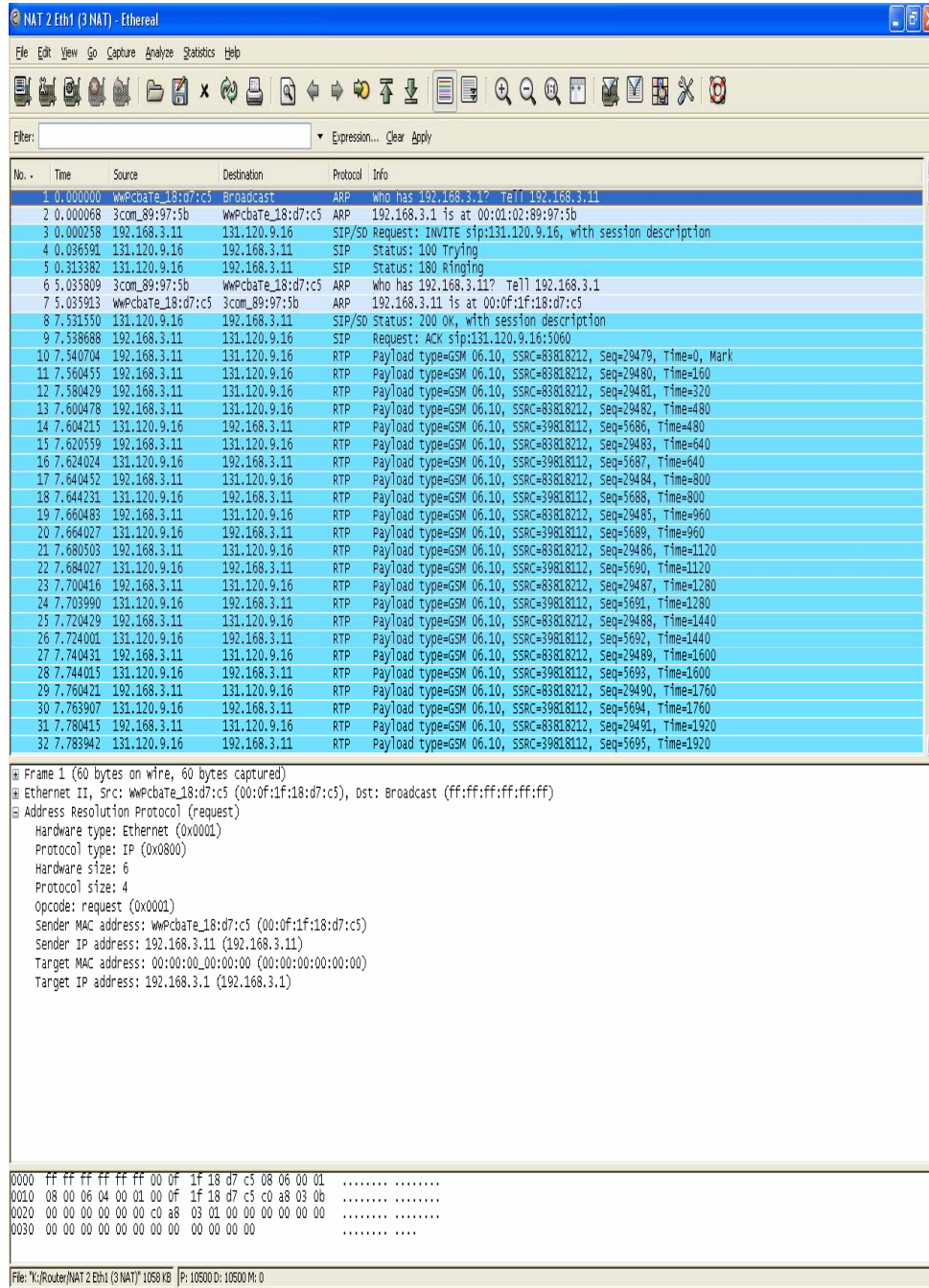


Figure 47. Test 5: Packet Capture on eth1 of NAT 2

E.5. Router Eth0

The following is a snapshot of the packets captured on the first interface (eth0) of Router.

The screenshot shows a Wireshark packet capture window titled "Router Eth1 (3 NAT) - Ethereal". The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Help), a toolbar with various icons, and a filter bar with the text "Filter: Expression... Clear Apply".

The main display area shows a list of 32 captured packets. The first packet is an ARP request from 192.168.202.11 to the broadcast address ff:ff:ff:ff:ff:ff. The subsequent packets are SIP messages, including an INVITE request and several RTP payloads of type GSM.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.202.11	Broadcast	ARP	who has 192.168.202.1? Tell 192.168.202.11
2	0.000054	192.168.202.11	192.168.202.11	ARP	192.168.202.1 is at 00:01:02:89:97:6f
3	0.000194	192.168.202.11	131.120.9.16	SIP/SDP	Request: INVITE sip:131.120.9.16, with session description
4	0.035790	131.120.9.16	192.168.202.11	SIP	Status: 100 Trying
5	0.312629	131.120.9.16	192.168.202.11	SIP	Status: 180 Ringing
6	5.034161	192.168.202.1	192.168.202.11	ARP	who has 192.168.202.11? Tell 192.168.202.1
7	5.034293	192.168.202.11	192.168.202.1	ARP	192.168.202.11 is at 00:e0:29:67:9e:9c
8	7.530799	131.120.9.16	192.168.202.11	SIP/SDP	Status: 200 OK, with session description
9	7.538277	192.168.202.11	131.120.9.16	SIP	Request: ACK sip:131.120.9.16:3060
10	7.540236	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29479, Time=0, Mark
11	7.559984	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29480, Time=160
12	7.579972	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29481, Time=320
13	7.600018	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29482, Time=480
14	7.603550	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5686, Time=480
15	7.620092	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29483, Time=640
16	7.623344	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5687, Time=640
17	7.639995	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29484, Time=800
18	7.643560	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5688, Time=800
19	7.660018	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29485, Time=960
20	7.663367	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5689, Time=960
21	7.680042	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29486, Time=1120
22	7.683369	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5690, Time=1120
23	7.699958	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29487, Time=1280
24	7.703320	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5691, Time=1280
25	7.719967	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29488, Time=1440
26	7.723327	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5692, Time=1440
27	7.739967	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29489, Time=1600
28	7.743349	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5693, Time=1600
29	7.759956	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29490, Time=1760
30	7.763249	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5694, Time=1760
31	7.779957	192.168.202.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29491, Time=1920
32	7.783272	131.120.9.16	192.168.202.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5695, Time=1920

Below the packet list, the details pane shows the structure of the first packet (Frame 1):

- Frame 1 (60 bytes on wire, 60 bytes captured)
- Ethernet II, Src: 192.168.202.11 (00:e0:29:67:9e:9c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
- Address Resolution Protocol (request)
 - Hardware type: Ethernet (0x0001)
 - Protocol type: IP (0x0800)
 - Hardware size: 6
 - Protocol size: 4
 - Opcode: request (0x0001)
 - Sender MAC address: 192.168.202.11 (00:e0:29:67:9e:9c)
 - Sender IP address: 192.168.202.11 (192.168.202.11)
 - Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
 - Target IP address: 192.168.202.1 (192.168.202.1)

The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII:

```
0000  ff ff ff ff ff ff 00 e0 29 67 9e 9c 08 06 00 01  ....Jg....
0010  08 00 06 04 00 01 00 e0 29 67 9e 9c c0 a8 ca 0b  ....Jg....
0020  00 00 00 00 00 00 c0 a8 ca 01 00 00 00 00 00 00  ....
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....
```

Figure 48. Test 5: Packet Capture on eth0 of Router

E.6. Router Eth1

The following is a snapshot of the packets captured on the second interface (eth1) of Router.

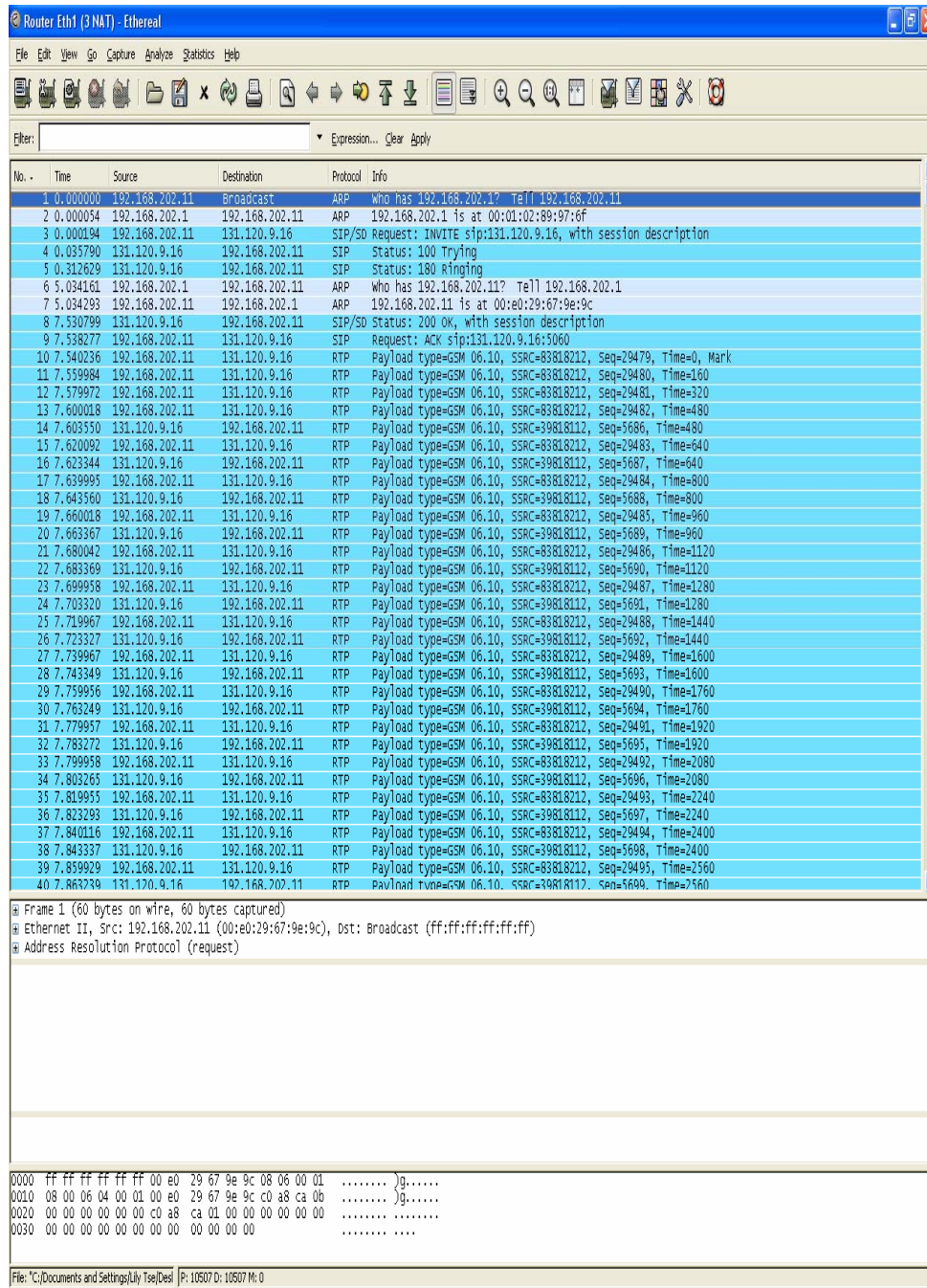


Figure 49. Test 5: Packet Capture on eth1 of Router

E.7. Router Eth2

The following is a snapshot of the packets captured on the third interface (eth2) of Router.

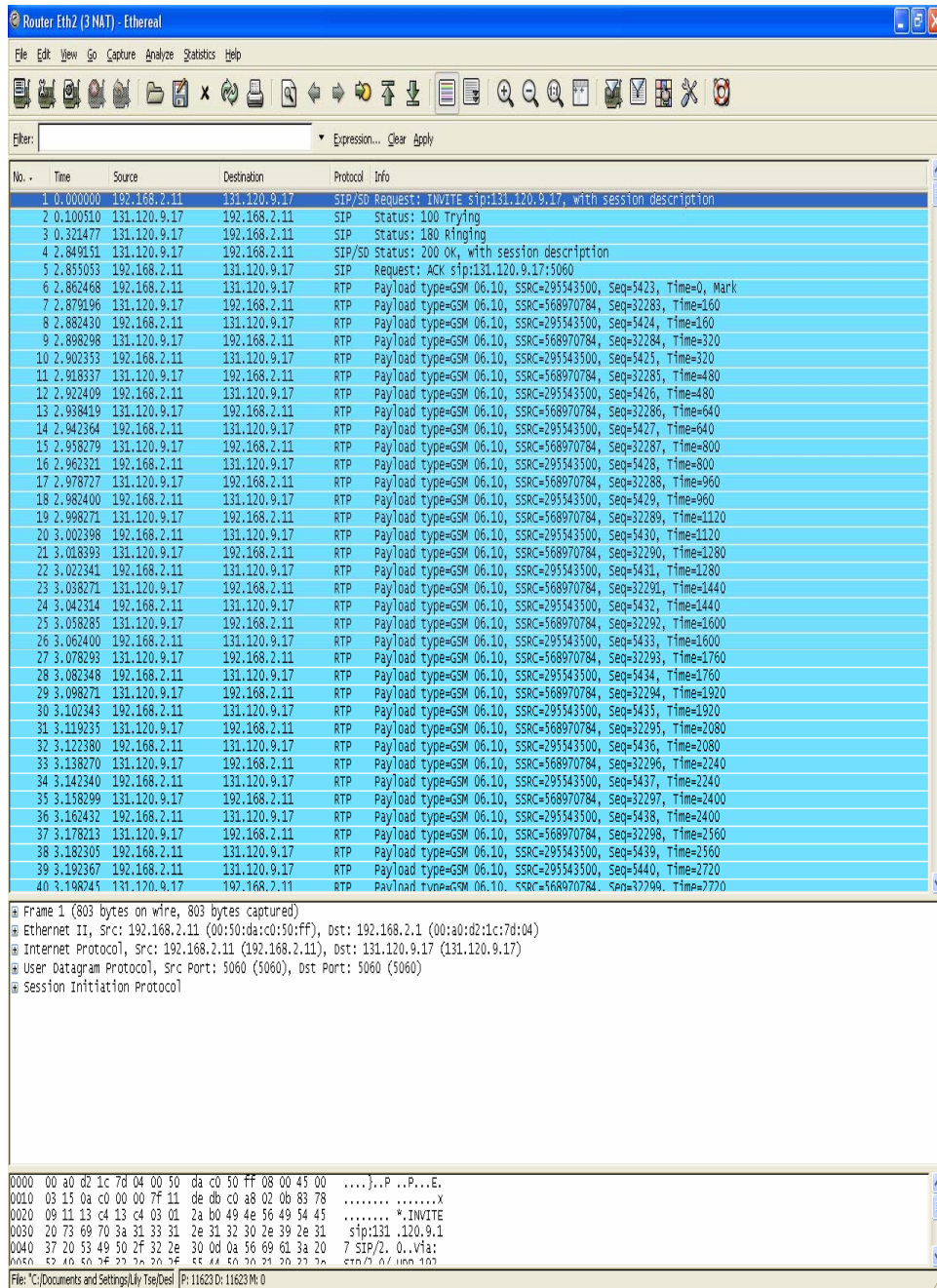
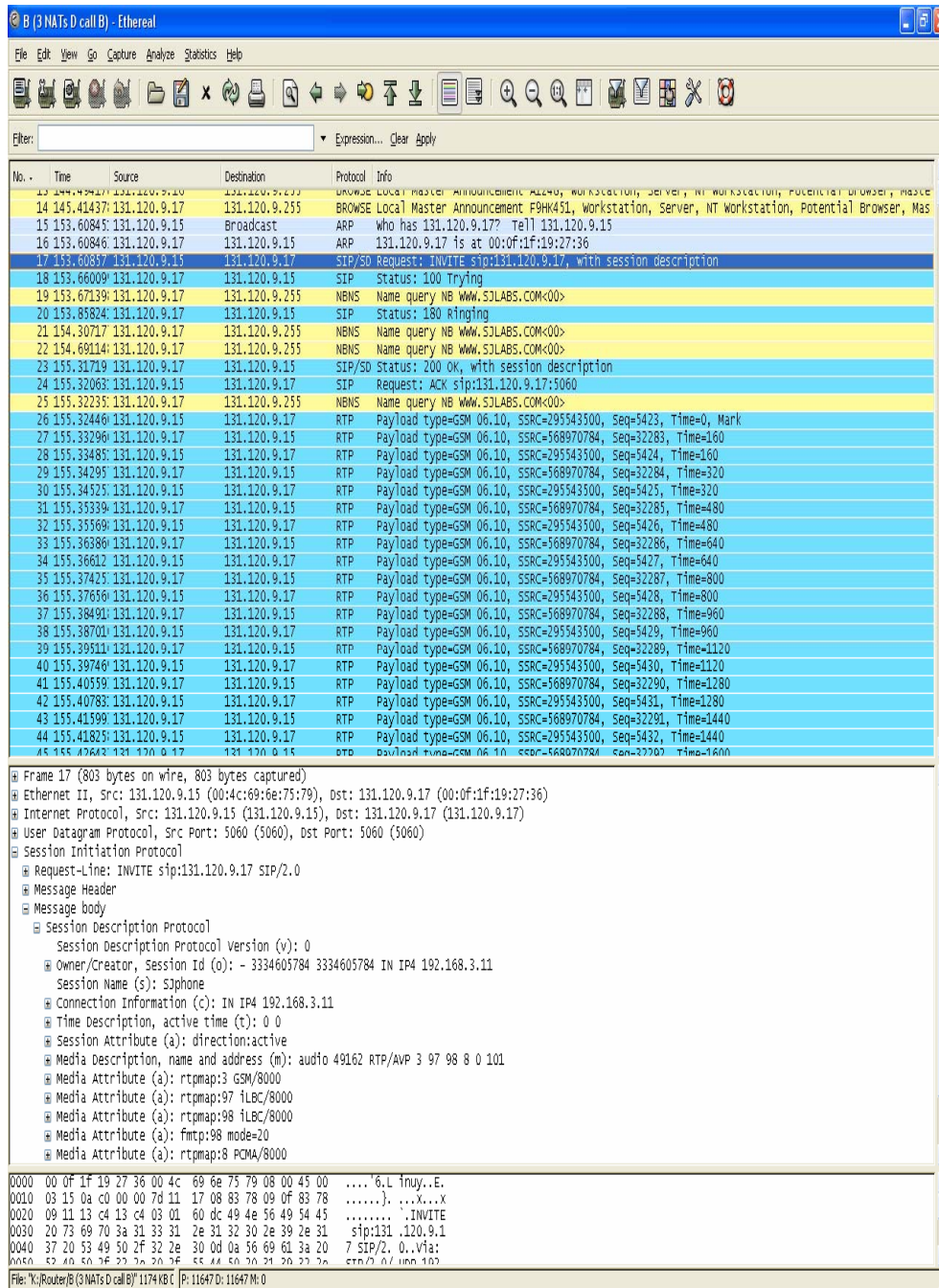


Figure 50. Test 5: Packet Capture on eth2 of Router

E.8. Client B

The following is a snapshot of the packets captured on Client B when it receives a call from Client D.



The screenshot shows a Wireshark packet capture on Client B (3 NATs D call B) - Ethereal. The packet list displays various protocols including BROWSE Local Master Announcement, ARP, SIP, NBNS, and RTP. The selected packet (17) is a SIP INVITE message. The packet details pane shows the Session Initiation Protocol structure, including the Request-Line, Message Header, and Message body. The message body contains a Session Description Protocol (SDP) offer with session name 'S3phone', connection information, and media description for audio and video.

No.	Time	Source	Destination	Protocol	Info
13	144.9417	131.120.9.10	131.120.9.17	BROWSE Local Master Announcement	ALL40, Workstation, Server, NT Workstation, Potential browser, Mas
14	145.41437	131.120.9.17	131.120.9.255	BROWSE Local Master Announcement	F9HK451, Workstation, Server, NT Workstation, Potential browser, Mas
15	153.60845	131.120.9.15	Broadcast	ARP	Who has 131.120.9.17? Tell 131.120.9.15
16	153.60846	131.120.9.17	131.120.9.15	ARP	131.120.9.17 is at 00:0f:1f:19:27:36
17	153.60857	131.120.9.15	131.120.9.17	SIP/SD	Request: INVITE sip:131.120.9.17, with session description
18	153.66009	131.120.9.17	131.120.9.15	SIP	Status: 100 Trying
19	153.67139	131.120.9.17	131.120.9.255	NBNS	Name query NB WWW.SJLABS.COM<00>
20	153.85824	131.120.9.17	131.120.9.15	SIP	Status: 180 Ringing
21	154.30717	131.120.9.17	131.120.9.255	NBNS	Name query NB WWW.SJLABS.COM<00>
22	154.69114	131.120.9.17	131.120.9.255	NBNS	Name query NB WWW.SJLABS.COM<00>
23	155.31719	131.120.9.17	131.120.9.15	SIP/SD	Status: 200 OK, with session description
24	155.32063	131.120.9.15	131.120.9.17	SIP	Request: ACK sip:131.120.9.17:5060
25	155.32235	131.120.9.17	131.120.9.255	NBNS	Name query NB WWW.SJLABS.COM<00>
26	155.32446	131.120.9.15	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5423, Time=0, Mark
27	155.33296	131.120.9.17	131.120.9.15	RTP	Payload type=GSM 06.10, SSRC=568970784, Seq=32283, Time=160
28	155.33485	131.120.9.15	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5424, Time=160
29	155.34295	131.120.9.17	131.120.9.15	RTP	Payload type=GSM 06.10, SSRC=568970784, Seq=32284, Time=320
30	155.34525	131.120.9.15	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5425, Time=320
31	155.35339	131.120.9.17	131.120.9.15	RTP	Payload type=GSM 06.10, SSRC=568970784, Seq=32285, Time=480
32	155.35569	131.120.9.15	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5426, Time=480
33	155.36386	131.120.9.17	131.120.9.15	RTP	Payload type=GSM 06.10, SSRC=568970784, Seq=32286, Time=640
34	155.36612	131.120.9.15	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5427, Time=640
35	155.37425	131.120.9.17	131.120.9.15	RTP	Payload type=GSM 06.10, SSRC=568970784, Seq=32287, Time=800
36	155.37656	131.120.9.15	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5428, Time=800
37	155.38491	131.120.9.17	131.120.9.15	RTP	Payload type=GSM 06.10, SSRC=568970784, Seq=32288, Time=960
38	155.38701	131.120.9.15	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5429, Time=960
39	155.39511	131.120.9.17	131.120.9.15	RTP	Payload type=GSM 06.10, SSRC=568970784, Seq=32289, Time=1120
40	155.39746	131.120.9.15	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5430, Time=1120
41	155.40559	131.120.9.17	131.120.9.15	RTP	Payload type=GSM 06.10, SSRC=568970784, Seq=32290, Time=1280
42	155.40783	131.120.9.15	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5431, Time=1280
43	155.41599	131.120.9.17	131.120.9.15	RTP	Payload type=GSM 06.10, SSRC=568970784, Seq=32291, Time=1440
44	155.41825	131.120.9.15	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5432, Time=1440
45	155.42642	131.120.9.17	131.120.9.15	RTP	Payload type=GSM 06.10, SSRC=568970784, Seq=32292, Time=1600

Frame 17 (803 bytes on wire (803 bytes captured))
Ethernet II, Src: 131.120.9.15 (00:4c:69:6e:75:79), Dst: 131.120.9.17 (00:0f:1f:19:27:36)
Internet Protocol, Src: 131.120.9.15 (131.120.9.15), Dst: 131.120.9.17 (131.120.9.17)
User Datagram Protocol, Src Port: 5060 (5060), Dst Port: 5060 (5060)
Session Initiation Protocol
Request-Line: INVITE sip:131.120.9.17 SIP/2.0
Message Header
Message body
Session Description Protocol
Session Description Protocol Version (v): 0
Owner/Creator, Session Id (o): - 3334605784 3334605784 IN IP4 192.168.3.11
Session Name (s): S3phone
Connection Information (c): IN IP4 192.168.3.11
Time Description, active time (t): 0 0
Session Attribute (a): direction:active
Media Description, name and address (m): audio 49162 RTP/AVP 3 97 98 8 0 101
Media Attribute (a): rtmap:3 GSM/8000
Media Attribute (a): rtmap:97 iLBC/8000
Media Attribute (a): rtmap:98 iLBC/8000
Media Attribute (a): fmp:98 mode=20
Media Attribute (a): rtmap:8 PCMA/8000

0000 00 0f 1f 19 27 36 00 4c 69 6e 75 79 08 00 45 006.L 1muy..E.
0010 03 15 0a c0 00 00 7d 11 17 08 83 78 09 0f 83 78}. ...X...x
0020 09 11 13 c4 13 c4 03 01 60 dc 49 4e 56 49 54 45INVITE
0030 20 73 69 70 3a 31 33 31 2e 31 32 30 2e 39 2e 31 sip:131.120.9.1
0040 37 20 53 49 50 2f 32 2e 30 0d 0a 56 69 61 3a 20 7 SIP/2.0..Via:
0050 52 40 50 7f 27 7a 20 7f 55 44 50 7a 21 20 27 7a .../ / / / / / / /

File: "K:\Router\B (3 NATs D call B)" 1174 KB C | P: 11647 D: 11647 M: 0

Figure 51. Test 5: Packet Capture on Client B

E.9. Client D

The following is a snapshot of the packets captured on Client D when it calls Client B.

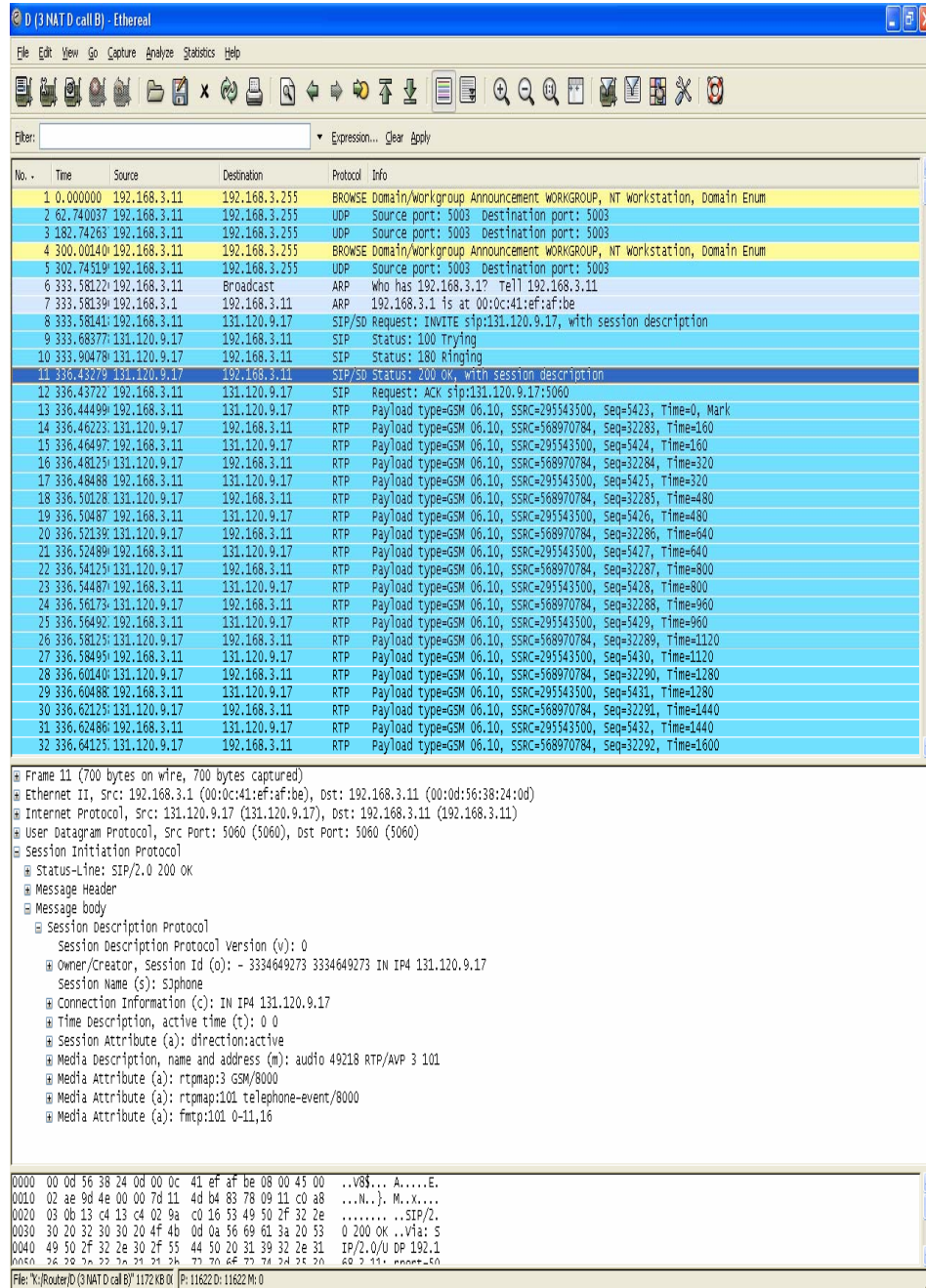


Figure 52. Test 5: Packet Capture on Client D

E.10. NAT 3 Eth0

The following is a snapshot of the packets captured on the first interface (eth0) of NAT 3.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.2.11	131.120.9.17	SIP/SD	Request: INVITE sip:131.120.9.17, with session description
2	0.101729	131.120.9.17	192.168.2.11	STP	Status: 100 Trying
3	0.322709	131.120.9.17	192.168.2.11	STP	Status: 180 Ringing
4	0.850369	131.120.9.17	192.168.2.11	SIP/SD	Status: 200 OK, with session description
5	2.855254	192.168.2.11	131.120.9.17	SIP	Request: ACK sip:131.120.9.17:5060
6	2.862946	192.168.2.11	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5423, Time=0, Mark
7	2.879909	131.120.9.17	192.168.2.11	RTP	Payload type=GSM 06.10, SSRC=368970784, Seq=32283, Time=160
8	2.882903	192.168.2.11	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5424, Time=160
9	2.899002	131.120.9.17	192.168.2.11	RTP	Payload type=GSM 06.10, SSRC=368970784, Seq=32284, Time=320
10	2.902806	192.168.2.11	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5425, Time=320
11	2.919041	131.120.9.17	192.168.2.11	RTP	Payload type=GSM 06.10, SSRC=368970784, Seq=32285, Time=480
12	2.922868	192.168.2.11	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5426, Time=480
13	2.939122	131.120.9.17	192.168.2.11	RTP	Payload type=GSM 06.10, SSRC=368970784, Seq=32286, Time=640
14	2.942834	192.168.2.11	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5427, Time=640
15	2.958976	131.120.9.17	192.168.2.11	RTP	Payload type=GSM 06.10, SSRC=368970784, Seq=32287, Time=800
16	2.962784	192.168.2.11	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5428, Time=800
17	2.979428	131.120.9.17	192.168.2.11	RTP	Payload type=GSM 06.10, SSRC=368970784, Seq=32288, Time=960
18	2.982859	192.168.2.11	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5429, Time=960
19	2.998972	131.120.9.17	192.168.2.11	RTP	Payload type=GSM 06.10, SSRC=368970784, Seq=32289, Time=1120
20	3.002860	192.168.2.11	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5430, Time=1120
21	3.019095	131.120.9.17	192.168.2.11	RTP	Payload type=GSM 06.10, SSRC=368970784, Seq=32290, Time=1280
22	3.022788	192.168.2.11	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5431, Time=1280
23	3.038971	131.120.9.17	192.168.2.11	RTP	Payload type=GSM 06.10, SSRC=368970784, Seq=32291, Time=1440
24	3.042776	192.168.2.11	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5432, Time=1440
25	3.058978	131.120.9.17	192.168.2.11	RTP	Payload type=GSM 06.10, SSRC=368970784, Seq=32292, Time=1600
26	3.062849	192.168.2.11	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5433, Time=1600
27	3.078988	131.120.9.17	192.168.2.11	RTP	Payload type=GSM 06.10, SSRC=368970784, Seq=32293, Time=1760
28	3.082807	192.168.2.11	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5434, Time=1760
29	3.098972	131.120.9.17	192.168.2.11	RTP	Payload type=GSM 06.10, SSRC=368970784, Seq=32294, Time=1920
30	3.102775	192.168.2.11	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5435, Time=1920
31	3.119935	131.120.9.17	192.168.2.11	RTP	Payload type=GSM 06.10, SSRC=368970784, Seq=32295, Time=2080
32	3.122827	192.168.2.11	131.120.9.17	RTP	Payload type=GSM 06.10, SSRC=295543500, Seq=5436, Time=2080

Frame 1 (803 bytes on wire, 803 bytes captured)

- Ethernet II, Src: 192.168.2.11 (00:50:da:c0:50:ff), Dst: 192.168.2.1 (00:a0:d2:1c:7d:04)
- Internet Protocol, Src: 192.168.2.11 (192.168.2.11), Dst: 131.120.9.17 (131.120.9.17)
- User Datagram Protocol, Src Port: 5060 (5060), Dst Port: 5060 (5060)
- Session Initiation Protocol
 - Request-Line: INVITE sip:131.120.9.17 SIP/2.0
 - Message Header
 - Message body
 - Session Description Protocol
 - Session Description Protocol version (v): 0
 - Owner/Creator, Session ID (o): - 3334605784 3334605784 IN IP4 192.168.3.11
 - Session Name (s): S3phone
 - Connection Information (c): IN IP4 192.168.3.11
 - Time Description, active time (t): 0 0
 - Session Attribute (a): direction:active
 - Media Description, name and address (m): audio 49162 RTP/AVP 3 97 98 8 0 101
 - Media Attribute (a): rtpmap:3 GSM/8000
 - Media Attribute (a): rtpmap:97 iLBC/8000
 - Media Attribute (a): rtpmap:98 iLBC/8000
 - Media Attribute (a): fmtp:98 mode=20
 - Media Attribute (a): rtpmap:8 PCMA/8000

0000 00 a0 d2 1c 7d 04 00 50 da c0 50 ff 08 00 45 00P...E.
 0010 03 15 0a c0 00 00 7f 11 de db c0 a8 02 0b 83 78X
 0020 09 11 13 c4 13 c4 03 01 2a b0 49 4e 56 49 54 45*.INVITE
 0030 20 73 69 70 3a 31 33 31 2e 31 32 30 2e 39 2e 31 sip:131.120.9.1
 0040 37 20 53 49 50 2f 32 2e 30 0d 0a 56 69 61 3a 20 7 SIP/2.0.via:
 0050 52 40 50 2f 27 2a 20 2f 55 44 50 20 21 20 27 2a 520 0 / via:101

File: "C:\Router\NAT 3 Eth0 (3 NAT) 1171 KB [P: 11623 D: 11623 M: 0

Figure 53. Test 5: Packet Capture on eth0 of NAT 3

E.11. NAT 3 Eth1

The following is a snapshot of the packets captured on the second interface (eth1) of NAT 3.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.3.11	Broadcast	ARP	Who has 192.168.3.1? Tell 192.168.3.11
2	0.000068	localhost-2. local	192.168.3.11	ARP	192.168.3.1 is at 00:01:02:89:97:5b
3	0.000258	192.168.3.11	131.120.9.16	SIP/SD	Request: INVITE sip:131.120.9.16, with session description
4	0.036591	131.120.9.16	192.168.3.11	SIP	Status: 100 Trying
5	0.313382	131.120.9.16	192.168.3.11	SIP	Status: 180 Ringing
6	5.035809	localhost-2. local	192.168.3.11	ARP	Who has 192.168.3.11? Tell 192.168.3.1
7	5.035913	192.168.3.11	localhost-2. local	ARP	192.168.3.11 is at 00:0f:1f:18:d7:c5
8	7.531550	131.120.9.16	192.168.3.11	SIP/SD	Status: 200 OK, with session description
9	7.538688	192.168.3.11	131.120.9.16	SIP	Request: ACK sip:131.120.9.16:5060
10	7.540704	192.168.3.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29479, Time=0, Mark
11	7.560455	192.168.3.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29480, Time=160
12	7.580429	192.168.3.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29481, Time=320
13	7.600478	192.168.3.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29482, Time=480
14	7.604215	131.120.9.16	192.168.3.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5686, Time=480
15	7.620559	192.168.3.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29483, Time=640
16	7.624024	131.120.9.16	192.168.3.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5687, Time=640
17	7.640452	192.168.3.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29484, Time=800
18	7.644231	131.120.9.16	192.168.3.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5688, Time=800
19	7.660483	192.168.3.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29485, Time=960
20	7.664027	131.120.9.16	192.168.3.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5689, Time=960
21	7.680503	192.168.3.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29486, Time=1120
22	7.684027	131.120.9.16	192.168.3.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5690, Time=1120
23	7.700416	192.168.3.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29487, Time=1280
24	7.703990	131.120.9.16	192.168.3.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5691, Time=1280
25	7.720429	192.168.3.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29488, Time=1440
26	7.724001	131.120.9.16	192.168.3.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5692, Time=1440
27	7.740431	192.168.3.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29489, Time=1600
28	7.744015	131.120.9.16	192.168.3.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5693, Time=1600
29	7.760421	192.168.3.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29490, Time=1760
30	7.763907	131.120.9.16	192.168.3.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5694, Time=1760
31	7.780415	192.168.3.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29491, Time=1920
32	7.783942	131.120.9.16	192.168.3.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5695, Time=1920
33	7.800415	192.168.3.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29492, Time=2080
34	7.803931	131.120.9.16	192.168.3.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5696, Time=2080
35	7.820416	192.168.3.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29493, Time=2240
36	7.823961	131.120.9.16	192.168.3.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5697, Time=2240
37	7.840581	192.168.3.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29494, Time=2400
38	7.843999	131.120.9.16	192.168.3.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5698, Time=2400
39	7.860400	192.168.3.11	131.120.9.16	RTP	Payload type=GSM 06.10, SSRC=83818212, Seq=29495, Time=2560
40	7.863907	131.120.9.16	192.168.3.11	RTP	Payload type=GSM 06.10, SSRC=39818112, Seq=5699, Time=2560

Frame 10476 (60 bytes on wire, 60 bytes captured)
 Ethernet II, Src: 192.168.3.11 (00:0f:1f:18:d7:c5), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Address Resolution Protocol (request)

```

0000  ff ff ff ff ff ff 0f 1f 18 d7 c5 08 06 00 01  .....
0010  08 00 06 04 00 01 00 0f 1f 18 d7 c5 c0 a8 03 06  .....
0020  00 00 00 00 00 00 c0 a8 03 01 00 00 00 00 00 00  .....
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
  
```

File: "C:\Documents and Settings\Jilly Tse\Desktop" P: 10500 D: 10500 M: 0

Figure 54. Test 5: Packet Capture on eth1 of NAT 3

E.12. NAT 1 Eth0

The following is a snapshot of the packets captured on the first interface (eth0) of NAT 1.

```
NAT1_ETH0 - WordPad
File Edit View Insert Format Help

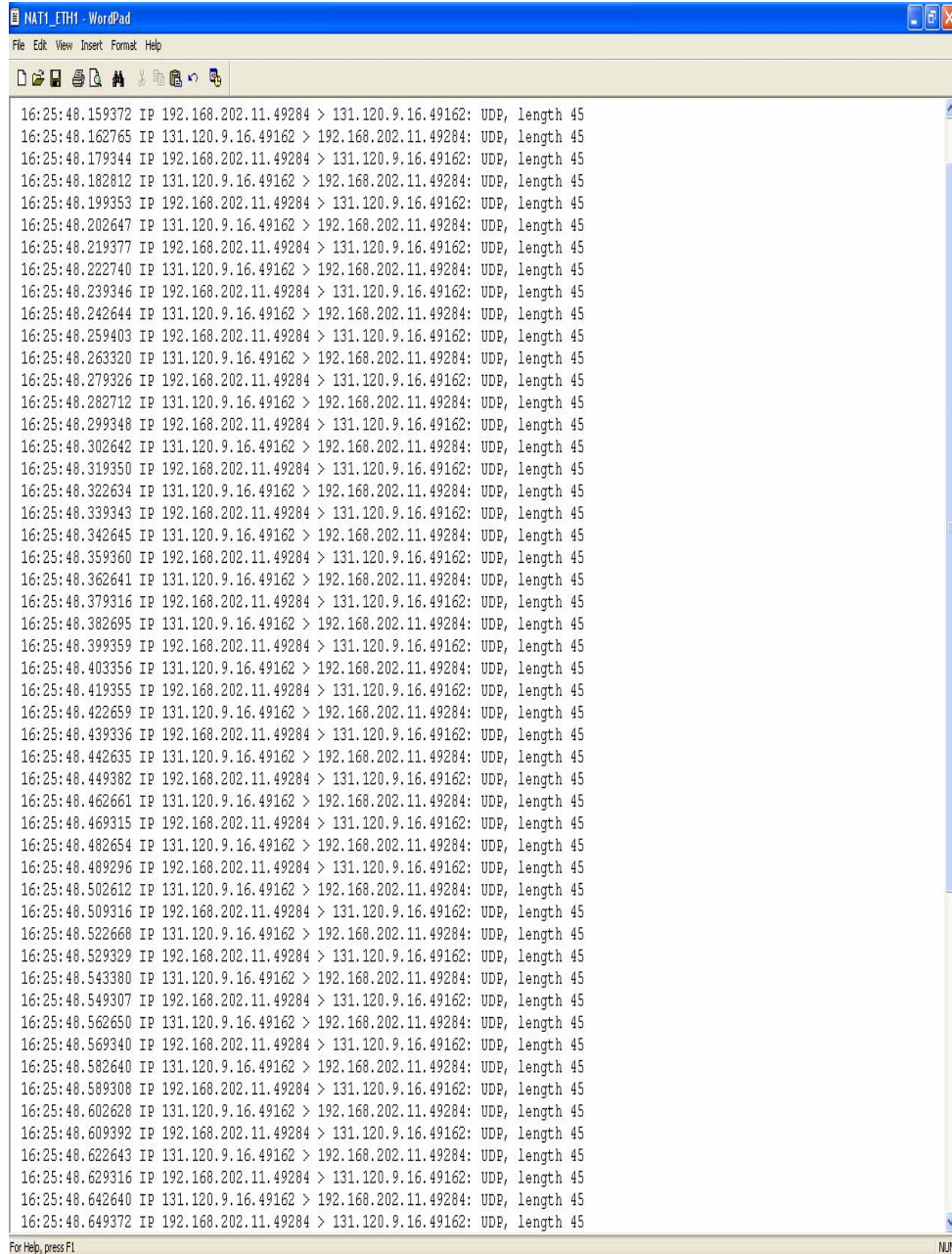
16:25:49.519295 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.532642 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.539430 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.552142 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.559306 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.572149 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.579251 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.582311 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.599409 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.602181 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.619294 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.622193 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.639268 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.642178 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.659367 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.662143 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.679254 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.682173 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.699316 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.702296 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.719246 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.722171 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.739302 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.742157 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.759326 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.762154 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.779284 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.782164 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.799300 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.802151 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.819233 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.822164 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.839311 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.842186 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.859399 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.862166 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.879374 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.882136 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.899275 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.902156 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.919290 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.922157 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.939367 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.942159 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.959288 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.962159 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.979303 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:49.982129 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:49.999288 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45
16:25:50.002150 IP 131.120.9.16.49162 > 131.120.9.15.49284: UDP, length 45
16:25:50.019301 IP 131.120.9.15.49284 > 131.120.9.16.49162: UDP, length 45

For Help, press F1
```

Figure 55. Test 5: Packet Capture on eth0 of NAT 1

E.13. NAT 1 Eth1

The following is a snapshot of the packets captured on the second interface (eth1) of NAT 1.



```
NAT1_ETH1 - WordPad
File Edit View Insert Format Help

16:25:48.159372 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.162765 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.179344 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.182812 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.199353 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.202647 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.219377 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.222740 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.239346 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.242644 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.259403 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.263320 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.279326 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.282712 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.299348 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.302642 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.319350 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.322634 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.339343 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.342645 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.359360 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.362641 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.379316 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.382695 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.399359 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.403356 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.419355 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.422659 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.439336 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.442635 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.449382 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.462661 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.469315 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.482654 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.489296 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.502612 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.509316 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.522668 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.529329 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.543380 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.549307 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.562650 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.569340 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.582640 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.589308 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.602628 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.609392 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.622643 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.629316 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45
16:25:48.642640 IP 131.120.9.16.49162 > 192.168.202.11.49284: UDP, length 45
16:25:48.649372 IP 192.168.202.11.49284 > 131.120.9.16.49162: UDP, length 45

For Help, press F1
```

Figure 56. Test 5: Packet Capture on eth1 of NAT 1

E.14. Analysis

As soon as Client C dialed the IP address of A, Client C sent out an “INVITE” message to Client A (red outline in Figure 44). The message had embedded SDP information to inform Client A that Client C would be sending and receiving RTP packets at 192.168.3.11 on port 49284 (purple outline in Figure 45). To acknowledge the invitation, Client A sent a “200 OK” packet to Client B with embedded SDP information indicating that it would send and receive RTP packets at 131.120.9.16 on port 49284 (green outline in Figure 44). The exchange of the “INVITE” and “200 OK” messages also occurred for the communication between Clients B and D. Client D informed Client B that it would send and receive RTP packets at 192.168.3.11 on 49162. On the other hand, Client B sent and received RTP packets at 131.120.9.17 on 49128. Figures 44 shows that Client A sent and received RTP packet directly to/from the public IP address of NAT 1. As explained in Appendix C, SJPhone will first attempt to send RTP media packets to the IP address indicated in the SDP messages (or the private IP address of Clients C and D). Since the firewall rules on Client A and Client B were configured to drop packets destined to the private IP address of Clients C and D, none of the initial packets sent out by Clients A or B could reach Clients C or D. Therefore, Clients A and B sent subsequent RTP packets to the IP address where it received Client C and D’s RTP media packets from (blue outline in Figure 44). The packet captures on Clients C indicate that the first RTP packet in the communication was sent by Client C. Even though NAT 1 was not explicitly configured to rewrite the destination IP address of incoming packets to 192.168.1.2 (public IP address of NAT 2), NAT 1 was able to intelligently determine this because *iptables* has a mechanism to maintain connection states of packets that are initiated from the local network. In our scenario, when the first RTP packet sent by Client C reached NAT 1, the packet was processed get changed from 192.168.202.11 to 192.168.120.9. At the same time, NAT 1 created an entry in its connection tracking table to store essential information (such as source and destination IP addresses and ports) that would allow it to associate incoming packets with Client C. This was also true for the communication between Clients D and B (refer to Figures 49 through 54).

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX G. TEST 6: MYSEA VOIP CONFIGURATION

The objective of Test 6 is to confirm that the MYSEA server could support simultaneous VoIP sessions from multiple MLS LAN clients. As described in Chapter IV, each network interface on the MLS server, currently an XTS-400, currently can only have one static route entry. However, two different static routes are needed to route packets destined for Client C and Client D. The test scenarios described here were intended to be workarounds to the XTS-400 routing problem. The goal was to forward packets to the Router if the packets were received on the single-level interface of the MLS server or forward packets to the NAT 1 if the packets were received on the MLS LAN interface of the MLS server. In other words, the goal was to configure XTS-400 to forward packets to its adjacent components, namely NAT 1 and Router, based on the network interface it receives the packets. Table 9 lists network configurations that were applied to the MLS server for the four test scenarios.

Test Scenario	Device Name	Gateway Address
1	/dev/ether0	192.168.0.27
	/dev/ether1	192.168.100.88
2	/dev/ether0	192.168.100.88
	/dev/ether1	192.168.0.27
3	/dev/ether1	192.168.100.88
	/dev/ether0	192.168.0.27
4	/dev/ether1	192.168.0.27
	/dev/ether0	192.168.100.88

Table 9. Test 6: Test Scenario Configurations

After each set of network configurations was applied to the MLS server, the Unix network utility *ping* was used to confirm the correctness of the network routing. In particular, each of the four IP addresses listed in Table 10 was pinged sequentially for four times from both the Router and NAT 1. Packets were captured using Ethereal at the MLS LAN interface of the Router (eth0, 192.168.0.27) and the single-level interface of NAT 1 (eth1, 192.168.100.88) for post-test analysis.

Interface	Device	IP Address
MLS LAN (eth0)	MLS server	192.168.0.130
single-level (eth1)	MLS server	192.168.100.130
single-level (eth1)	NAT 1	192.168.100.88
public (eth0)	NAT 1	131.120.9.15

Table 10. Test 6: Ping Operations

None of the four tests was completed successfully, i.e., there was at least one interface on the MLS server and/or the public NAT that the Router or NAT 1 was unable to ping. See the next four sections for the results.

A. Network Topology

Refer to Figure 14 and Figure 15 for the physical and logical network topology. Note that the clients and NAT 3 were not used in the test scenarios.

B. Equipment Requirements

B.1. NAT 1, NAT 2 and Router

- B.1.1. Linux Operating System (Fedora Core 4)
- B.1.2. netfilter and iptables
- B.1.3. Ethereal
- B.1.4. Two network cards (for NAT 1 and NAT 2)
- B.1.5. Three network cards (for Router only)

B.2. MLS Server

- B.2.1. XTS-400

B.3. Additional Equipment

- B.3.1. Cross-over cables and a switch or hub to implement the network architecture illustrated in Figure 14.

C. Installation and Configuration

C.1. MLS Server

- C.1.1. Configure two network interfaces to be at the same level as the MLS LAN by entering:
min as the security level

max as the integrity level

C.2. Router

- C.2.1. Configure eth0 by editing `/etc/sysconfig/network-scripts/ifcfg-eth0` to include the following:

```
DEVICE=eth0
BOOTPROTO=NONE
IPADDR=192.168.0.27
NETMASK=255.255.255.0
GATEWAY=192.168.0.130
```

- C.2.2. Activate eth0 by running:

```
ifup eth0
```

- C.2.3. Configure eth1 by editing `/etc/sysconfig/network-scripts/ifcfg-eth1` to include the following:

```
DEVICE=eth1
BOOTPROTO=NONE
IPADDR=192.168.202.1
NETMASK=255.255.255.0
```

- C.2.4. Activate eth1 by running:

```
ifup eth1
```

- C.2.5. Configure eth2 by editing and saving `/etc/sysconfig/network-scripts/ifcfg-eth2` to include the following:

```
DEVICE=eth2
BOOTPROTO=NONE
IPADDR=192.168.2.1
NETMASK=255.255.255.0
```

- C.2.6. Activate eth2 by running:

```
ifup eth2
```

- C.2.7. Enable IP Forwarding by running:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

- C.2.8. Flush any existing firewall and NAT rules by running:

```
iptables -F
```

```
iptables -t nat -F
```

C.3. NAT 1

- C.3.1. Configure eth0 by editing `/etc/sysconfig/network-scripts/ifcfg-eth0` to include the following:

```
DEVICE=eth0
BOOTPROTO=NONE
IPADDR=131.120.9.15
NETMASK=255.255.255.0
GATEWAY=131.120.9.17
```

- C.3.2. Activate eth0 by running:

```
ifup eth0
```

- C.3.3. Configure eth1 by editing and saving `/etc/sysconfig/network-scripts/ifcfg-eth1` to include the following:

```
DEVICE=eth1
BOOTPROTO=NONE
IPADDR=192.168.100.88
NETMASK=255.255.255.0
```

- C.3.4. Activate eth1 by running:

```
ifup eth1
```

- C.3.5. Configure static routes by running:

```
route add -net 192.168.202.0 netmask 255.255.255.0 gw 192.168.100.130 eth1
route add -net 192.168.2.0 netmask 255.255.255.0 gw 192.168.100.130 eth1
route add -net 192.168.0.0 netmask 255.255.255.0 gw 192.168.100.130 eth1
```

- C.3.6. Enable IP Forwarding by running:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

- C.3.7. Flush any existing firewall and NAT rules by running:

```
iptables -F
iptables -t nat -F
```

- C.3.8. Configure NAT rule by running:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 131.120.9.15
```

C.4. NAT 2

C.4.1. Configure eth0 by editing and saving `/etc/sysconfig/network-scripts/ifcfg-eth0` to include the following:

```
DEVICE=eth0
BOOTPROTO=NONE
IPADDR=196.168.202.11
NETMASK=255.255.255.0
GATEWAY=198.168.202.1
```

C.4.2. Activate eth0 by running:

```
ifup eth0
```

C.4.3. Configure eth1 by editing and saving `/etc/sysconfig/network-scripts/ifcfg-eth1` to include the following:

```
DEVICE=eth1
BOOTPROTO=NONE
IPADDR=192.168.3.1
NETMASK=255.255.255.0
```

C.4.4. Activate eth1 by running:

```
ifup eth1
```

C.4.5. Enable IP Forwarding by running:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

C.4.6. Flush any existing firewall and NAT rules by running:

```
iptables -F
iptables -t nat -F
```

C.4.7. Configure NAT rules by running:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 192.168.202.11
iptables -t nat -A PREROUTING -i eth0 -j DNAT --to 192.168.3.11
```

D. Scenario 1

D.1. Description

The MLS server is configured in order as follows: any packets received on its MLS LAN (eth0, 192.168.0.130) is forwarded to the MLS LAN interface (eth0, 192.168.0.27) of the Router and any packets received on its single-level (eth1,

192.168.100.130) is forwarded to the single-level interface (eth1, 192.168.100.88) of NAT 1.

D.2. Operations

First, NAT 2 pings:

1. eth0 of MLS server
2. eth1 of MLS server
3. eth1 of NAT 1
4. eth0 of NAT 1

Then, Router pings:

5. eth0 of MLS server
6. eth1 of MLS server
7. eth1 of NAT 1
8. eth0 of NAT 1

D.3. Network Configuration on MLS Server

D.3.1. Type the following answers when prompted:

SAK

Enter command? tcpip_edit

Enter editor request? add

Enter TCP/IP daemon name? tcpip_mls

Enter TCPIP/IP daemon description? TCP/IP for MLS LAN network

Enter domain name? cisrlabmlstestbed1.com

Enter host name? mlsserver

Enable the subnets local flag? n

Enable the IP forwarding flag? y

Enable the IP send redirect flag? y

Enable the shutdown on failure flag? n

Use default TCP maximum retransmission? y

Add the network interface configuration? y

Enter TCP/IP device name? /dev/ether0

Enter interface address? 192.168.0.130

Enter destination address?	0.0.0.0
Enter broadcast address?	192.168.0.255
Enter network mask?	255.255.255.0
Add another network interface entry?	y
Enter TCP/IP device name?	/dev/ether1
Enter interface address?	192.168.100.130
Enter destination address?	0.0.0.0
Enter broadcast address?	192.168.100.255
Enter network mask?	255.255.255.0
Add another network interface entry?	n
Add the route configuration?	y
Enter TCP/IP device name	/dev/ether0
Is this route a default route	n
Enter destination address	0.0.0.0
Is destination address a host	n
Enter gateway address	192.168.0.27
Enter route metric	1
Add another network route entry	y
Enter TCP/IP device name	/dev/ether1
Is this route a default route	n
Enter destination address	0.0.0.0
Is destination address a host	n
Enter gateway address	192.168.100.88
Enter route metric	1
Add another network route entry	n
Add the resolver configuration?	n

D.4.Preparation and Testing

D.4.1. On NAT 1,

D.4.1.1. Launch Ethereal

D.4.1.2. Go to the **Capture** menu

D.4.1.3. Go to **Interfaces**

D.4.1.4. Click on **Capture 192.168.100.88**

D.4.2. On Router,

D.4.2.1. Launch **Ethereal**

D.4.2.2. Go to the **Capture** menu

D.4.2.3. Go to **Interfaces**

D.4.2.4. Click on **Capture 192.168.0.27**

D.4.3. On NAT 2,

D.4.3.1. Run the following commands:

ping -c 4 192.168.0.130

ping -c 4 192.168.100.130

ping -c 4 192.168.100.88

ping -c 4 131.120.9.15

D.4.4. Repeat the above commands on the Router

D.4.5. Stop Ethereal captures on both NAT 1 and Router

D.5.Result

Table 11 lists the result of the Scenario 1. The first column shows where the ping was initiated and the first row shows what hosts/IP addresses were pinged. Neither NAT 2 nor the Router was able to ping the public interface of the Public NAT.

to from	192.168.0.130 (eth0, MLS LAN interface of MLS server)	192.168.100.130 (eth1, single-level interface of MLS server)	192.168.100.88 (eth1, single-level interface of NAT 1)	131.120.9.15 (eth0, public interface of NAT 1)
NAT 2	Successful	Successful	Successful	Failed
Router	Successful	Successful	Successful	Failed

Table 11. Test 6: Scenario 1 Result

D.6. Packet Capture when pinged from NAT 2

D.6.1. Router

The following two figures are snapshots of the packets captured on eth0 of Router when the four interfaces were pinged from NAT 2.

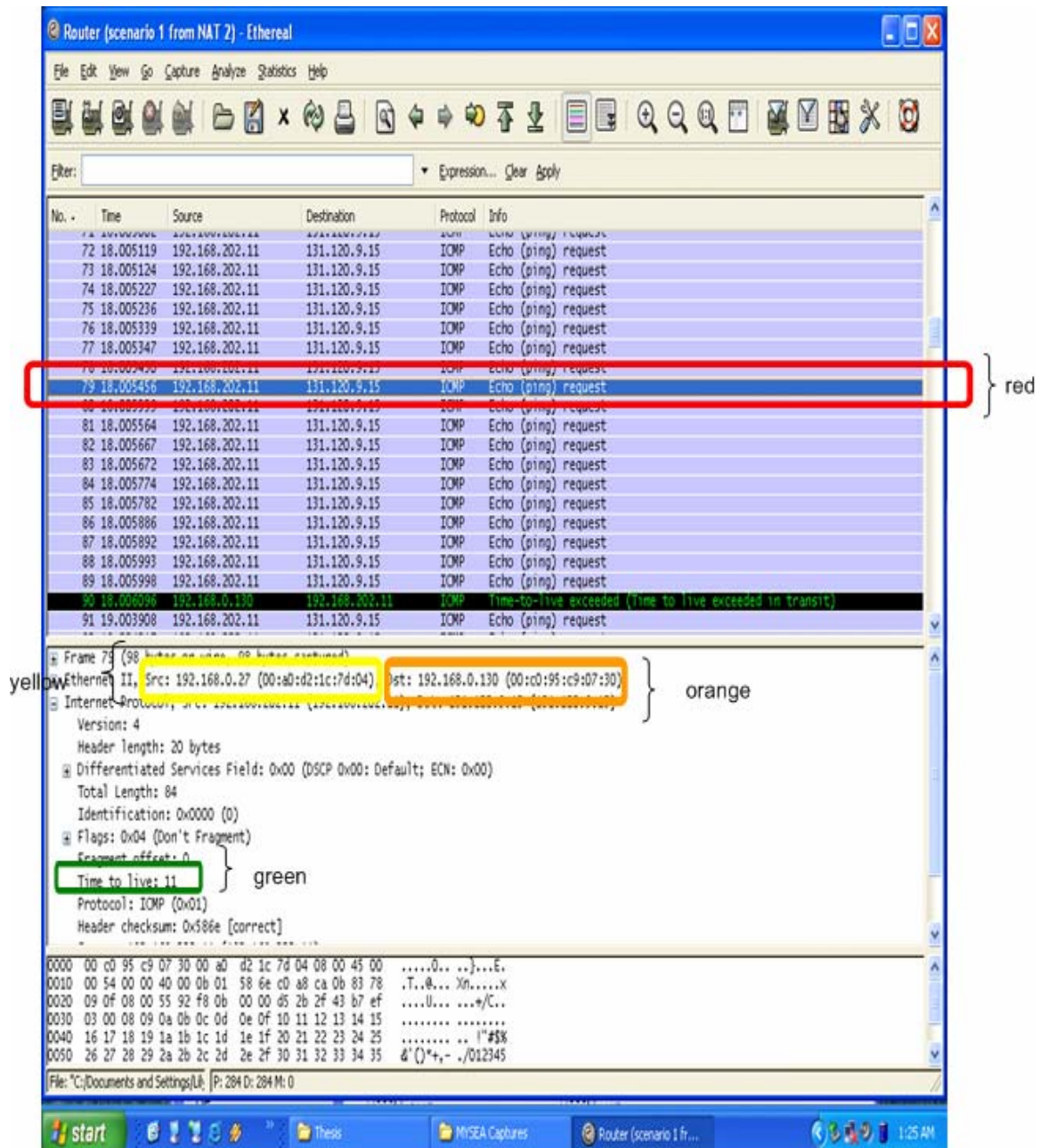


Figure 57. Test 6: Scenario 1 Packet Capture on Router (pinged from NAT 2), Part 1

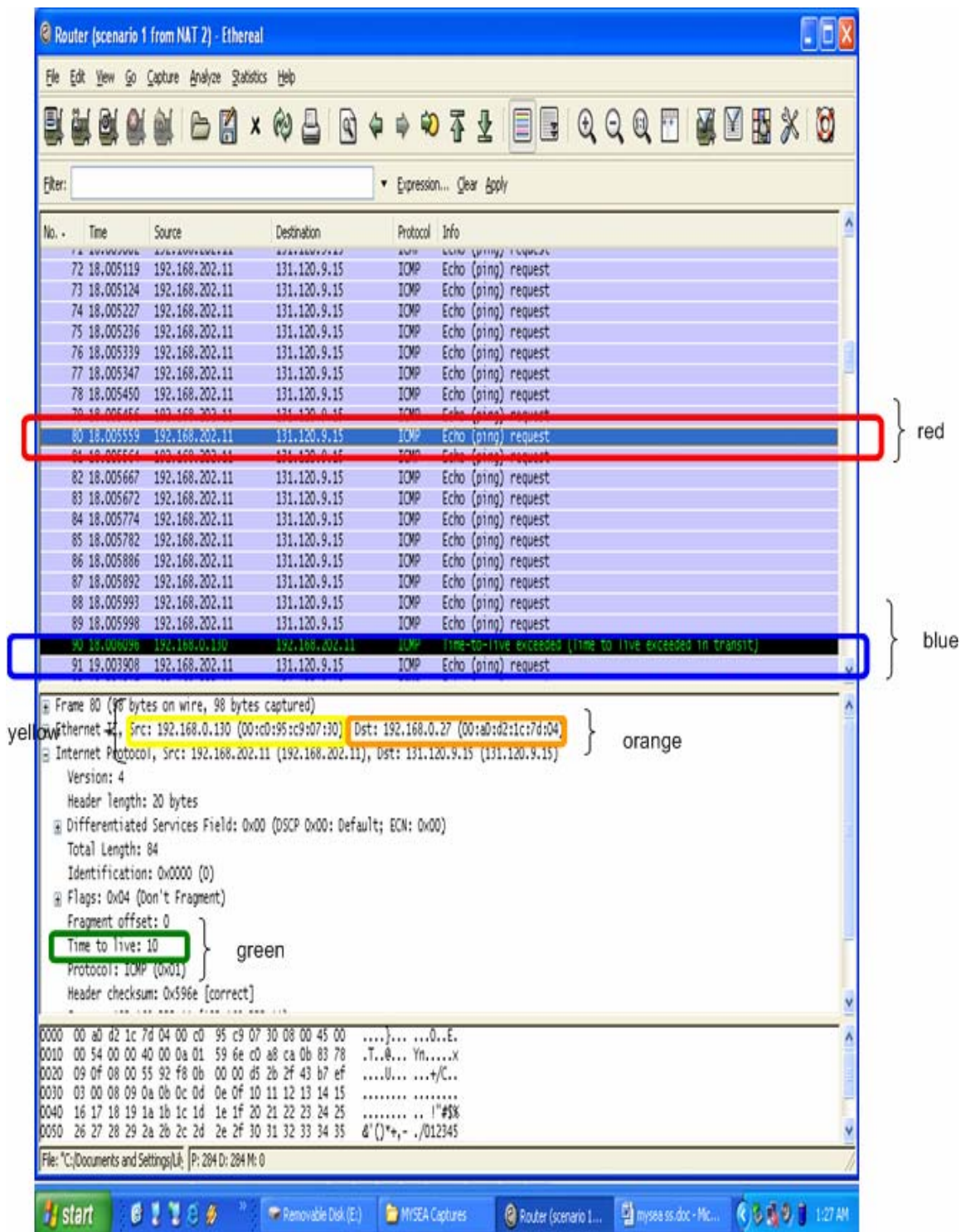


Figure 58. Test 6: Scenario 1 Packet Capture on Router (pinged from NAT 2), Part 2

D.6.2. NAT 1

The following is a snapshot of the packets captured on the eth1 of NAT 1 when the four interfaces were pinged from NAT 2.

The screenshot shows a Wireshark capture titled "NAT 1 (scenario 1 from NAT 2) - Ethereal". The packet list contains 12 entries. The first two are ARP requests and replies between 192.168.100.130 and 192.168.100.88. The next eight are ICMP Echo (ping) requests and replies between 192.168.202.11 and 192.168.100.88. The last two are ARP requests and replies between 192.168.100.88 and 192.168.100.130.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.100.130	broadcast	ARP	Who has 192.168.100.88? Tell 192.168.100.130
2	0.000084	192.168.100.88	192.168.100.130	ARP	192.168.100.88 is at 00:c0:95:c9:07:31
3	0.000123	192.168.202.11	192.168.100.88	ICMP	Echo (ping) request
4	0.000168	192.168.100.88	192.168.202.11	ICMP	Echo (ping) reply
5	1.000749	192.168.202.11	192.168.100.88	ICMP	Echo (ping) request
6	1.000841	192.168.100.88	192.168.202.11	ICMP	Echo (ping) reply
7	2.000597	192.168.202.11	192.168.100.88	ICMP	Echo (ping) request
8	2.000682	192.168.100.88	192.168.202.11	ICMP	Echo (ping) reply
9	3.000448	192.168.202.11	192.168.100.88	ICMP	Echo (ping) request
10	3.000534	192.168.100.88	192.168.202.11	ICMP	Echo (ping) reply
11	4.998951	192.168.100.88	192.168.100.130	ARP	Who has 192.168.100.130? Tell 192.168.100.88
12	4.999041	192.168.100.130	192.168.100.88	ARP	192.168.100.130 is at 00:c0:95:c9:07:31

Frame 1 (60 bytes on wire, 60 bytes captured)
 Ethernet II, Src: 192.168.100.130 (00:c0:95:c9:07:31), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Address Resolution Protocol (request)

```

0000  ff ff ff ff ff ff 00 c0 95 c9 07 31 08 06 00 01  .....I....
0010  08 00 06 04 00 01 00 c0 95 c9 07 31 c0 a8 64 82  .....l..d.
0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....M.....
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
  
```

File: E:\NAT 1 (scenario 1 from NAT 2) [P: 12 D: 12 M: 0]

Figure 59. Test 6: Scenario 1 Packet Capture on NAT 1 (pinged from NAT 2)

D.6.3. Analysis

NAT 2 was able to ping 192.168.0.130 and 192.168.100.130 because the MLS server shared a peer-to-peer relationship with the Router and the Router had logic to route packets between the MLS server and NAT 2. It was also able to ping 192.168.100.88 because XTS-400 was capable of routing the Echo requests and replies to its immediate peers that in turn, routed the packets to the destination. Pinging 131.120.9.15 was unsuccessful from NAT 2. The Router saw ICMP requests when NAT 2 pinged 131.120.9.15 (red outline in Figure 57). The ICMP requests were routed from 192.168.0.27 (yellow outline in Figure 57) to 192.168.0.130 (orange outline in Figure 57). As soon as the MLS server received the requests, the XTS-400 bounced them back to the IP address from which they came (yellow and orange outlines in Figure 58). Note that the time-to-live field was decremented from 11 (green outline in Figure 57) to 10 (green outline in Figure 58) indicating that the two requests seen in the packet capture were, in fact, the same packet. This sequence of events continued until the time-to-live was exceeded in transit (blue outline in Figure 57). Thus, the ICMP was never able to reach NAT 1 as shown in Figure 59.

D.7. Packet Capture when pinged from Router

D.7.1. Router

The following two pictures are snapshots of the packets captured on eth0 of the Router when the four interfaces were pinged from Router.

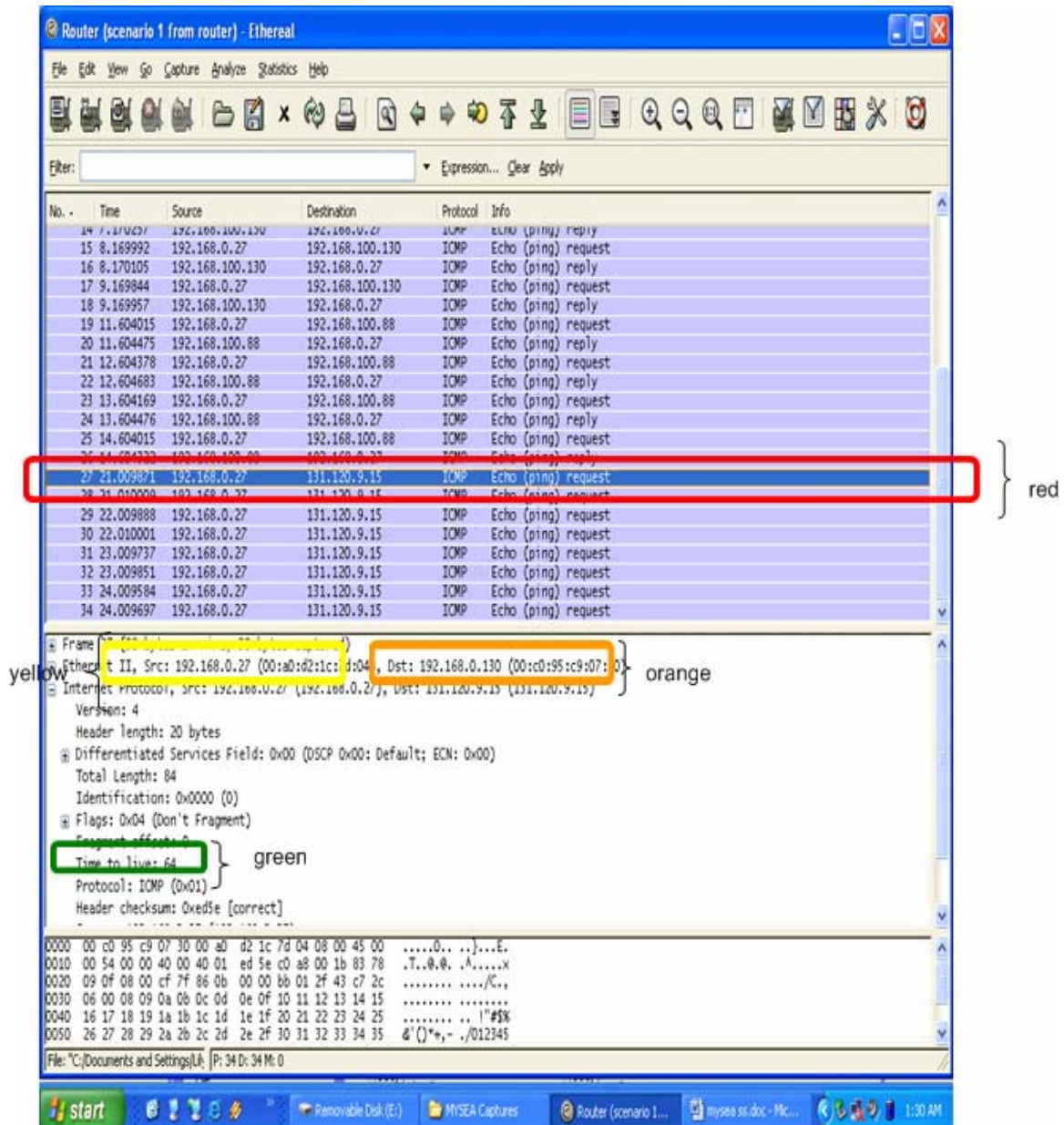


Figure 60. Test 6: Scenario 1 Packet Capture on Router (pinged from Router), Part 1

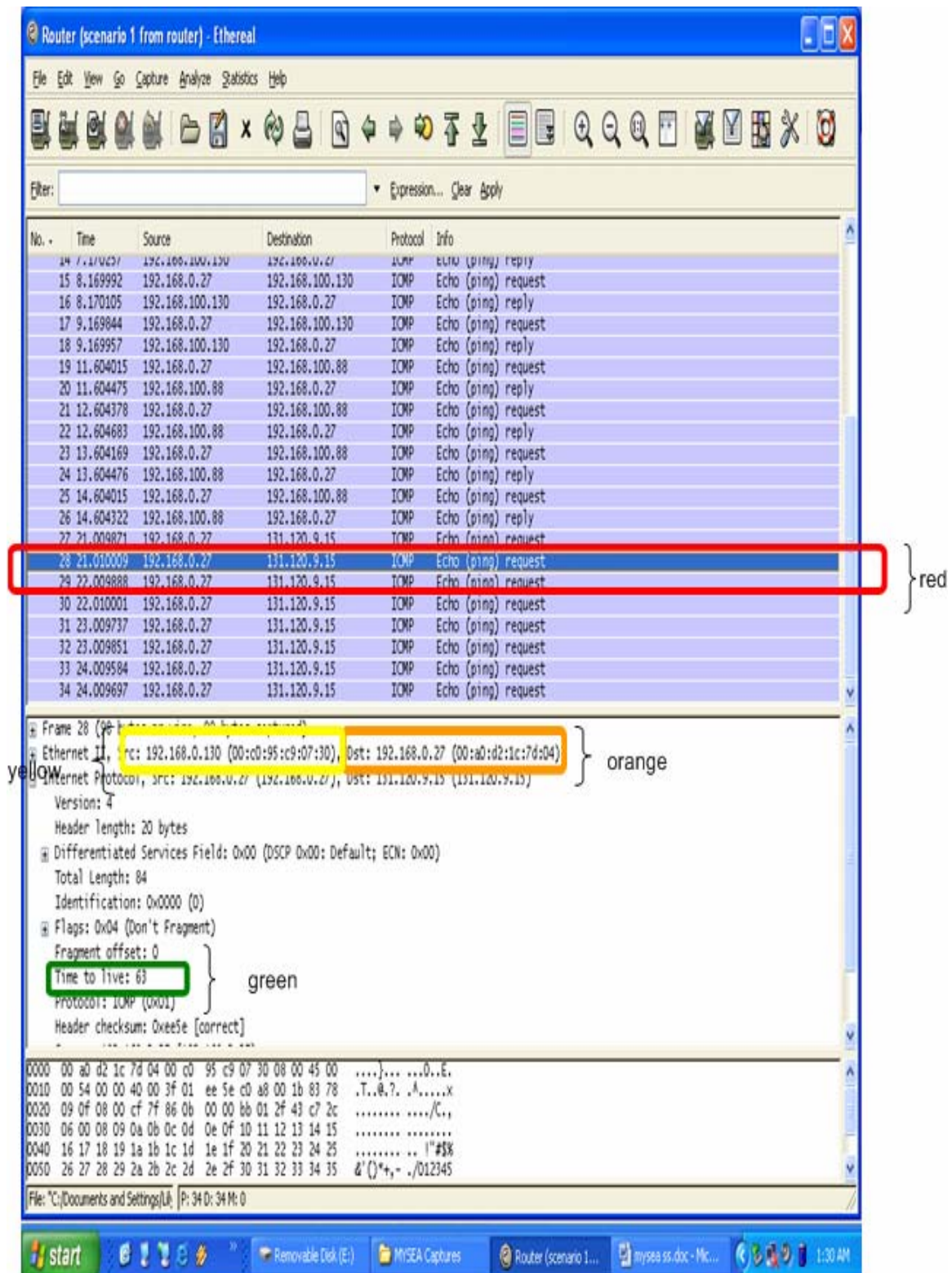


Figure 61. Test 6: Scenario 1 Packet Capture on Router (pinged from Router), Part 2

D.7.2. NAT 1

The following is a snapshot of the packets captured on the NAT 1 when the four interfaces were pinged from Router.

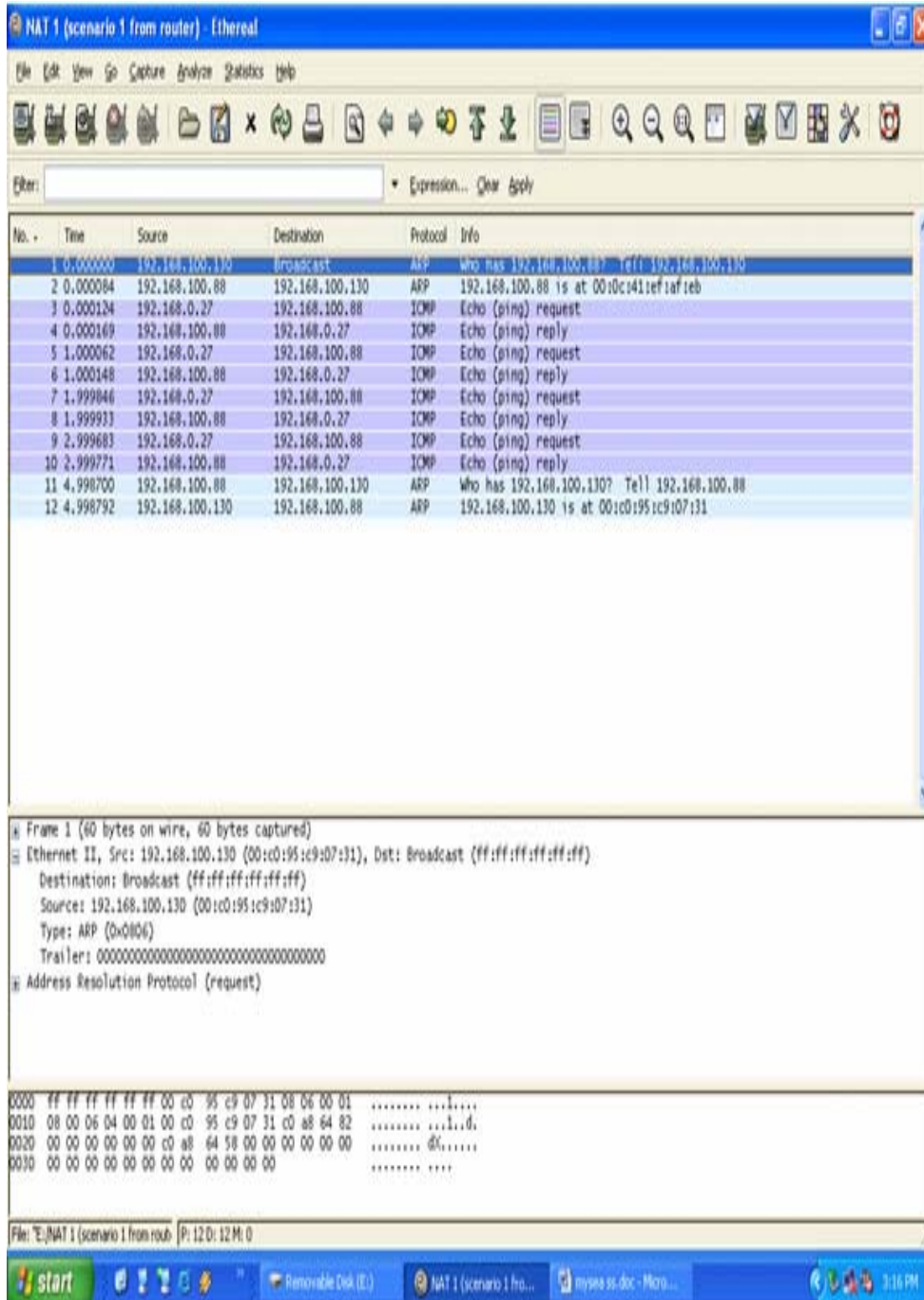


Figure 62. Test 6: Scenario 1 Packet Capture on NAT 1 (pinged from Router)

D.7.3. Analysis

Figures 60 to 62 indicate similar behaviors when the four interfaces were pinged from the Router instead of NAT 2. The Router could ping 192.168.0.130, 192.168.100.130 and 192.168.100.88 for the reason explained in D.6.3. Pinging 131.120.9.15 from the Router had a different behavior than the one seen when it was pinged from NAT 2 such that the Echo requests never had their time-to-live exceeded. Each of the four Echo requests was routed between 192.168.0.27 and 192.168.0.130 twice (yellow and orange outlines in Figure 60 and Figure 61). Each request was routed from 192.168.0.27 first and then XTS-400 routed it back to 192.168.0.130 (Router) as soon as XTS-400 received it. The decrement in the time-to-live field indicated that the same request was routed back and forth (green outlines in Figure 60 and Figure 61). The request was not routed further when the Router received it because the Router recognized its own packet and thus, stopped routing it. As a result, NAT 1 never saw the Echo requests as shown in Figure 62.

E. Scenario 2

E.1. Description

The MLS server is configured in order as follows: any packets received on its MLS LAN interface (eth0, 192.168.0.130) is forwarded to the single-level interface of NAT 1 (eth1, 192.168.100.88) and any packets received on its single-level interface (eth1, 192.168.100.130) is forwarded to the public interface of Router (eth0, 192.168.0.27).

E.2. Operations

First, NAT 2 pings:

1. eth0 of MLS server
2. eth1 of MLS server
3. eth1 of NAT 1
4. eth0 of NAT 1

Then, Router pings:

5. eth0 of MLS server
6. eth1 of MLS server
7. eth1 of NAT 1

8. eth0 of NAT 1

E.3. Network Configuration on MLS Server

E.3.1. Type the following answers when prompted:

SAK

Enter command?	tcpip_edit
Enter editor request?	add
Enter TCP/IP daemon name?	tcpip_mls
Enter TCPIP/IP daemon description?	TCP/IP for MLS LAN

network

Enter domain name?	cisrlabmlstestbed1.com
Enter host name?	mlsserver
Enable the subnets local flag?	n
Enable the IP forwarding flag?	y
Enable the IP send redirect flag?	y
Enable the shutdown on failure flag?	n
Use default TCP maximum retransmission?	y
Add the network interface configuration?	y
Enter TCP/IP device name?	/dev/ether0
Enter interface address?	192.168.0.130
Enter destination address?	0.0.0.0
Enter broadcast address?	192.168.0.255
Enter network mask?	255.255.255.0
Add another network interface entry?	y
Enter TCP/IP device name?	/dev/ether1
Enter interface address?	192.168.100.130
Enter destination address?	0.0.0.0
Enter broadcast address?	192.168.100.255
Enter network mask?	255.255.255.0
Add another network interface entry?	n
Add the route configuration?	y
Enter TCP/IP device name	/dev/ether0

Is this route a default route	n
Enter destination address	0.0.0.0
Is destination address a host	n
Enter gateway address	192.168.100.88
Enter route metric	1
Add another network route entry	y
Enter TCP/IP device name	/dev/ether1
Is this route a default route	n
Enter destination address	0.0.0.0
Is destination address a host	n
Enter gateway address	192.168.0.27
Enter route metric	1
Add another network route entry	n
Add the resolver configuration?	n

E.4. Preparation and Testing

E.4.1. On NAT 1,

- E.4.1.1. Launch Ethereal
- E.4.1.2. Go to the **Capture** menu
- E.4.1.3. Go to **Interfaces**
- E.4.1.4. Click on **Capture 192.168.100.88**

E.4.2. On Router,

- E.4.2.1. Launch Ethereal
- E.4.2.2. Go to the **Capture** menu
- E.4.2.3. Go to **Interfaces**
- E.4.2.4. Click on **Capture 192.168.0.27**

E.4.3. On NAT 2,

E.4.3.1. Run the following commands:

```
ping -c 4 192.168.0.130
ping -c 4 192.168.100.130
ping -c 4 192.168.100.88
ping -c 4 131.120.9.15
```

E.4.4. Repeat the above commands on the Router

E.4.5. Stop Ethereal captures on both NAT 1 and Router

E.5. Result

Table 12 lists the result of the Scenario 2. The first column shows where the ping was initiated and the first row shows what hosts/IP addresses were pinged. NAT 2 was unable to ping any of the four IP addresses.

to from	192.168.0.130 (eth0, MLS LAN interface of MLS server)	192.168.100.130 (eth1, single-level interface of MLS server)	192.168.100.88 (eth1, single-level interface of NAT 1)	131.120.9.15 (eth0, public interface of NAT 1)
NAT 2	Failed	Failed	Failed	Failed
Router	Successful	Successful	Successful	Successful

Table 12. Test 6: Scenario 2 Result

E.6. Packet Capture when pinged from NAT 2

E.6.1. Router

The following is a snapshot of the packets captured on the Router when the four interfaces were pinged from NAT 2.

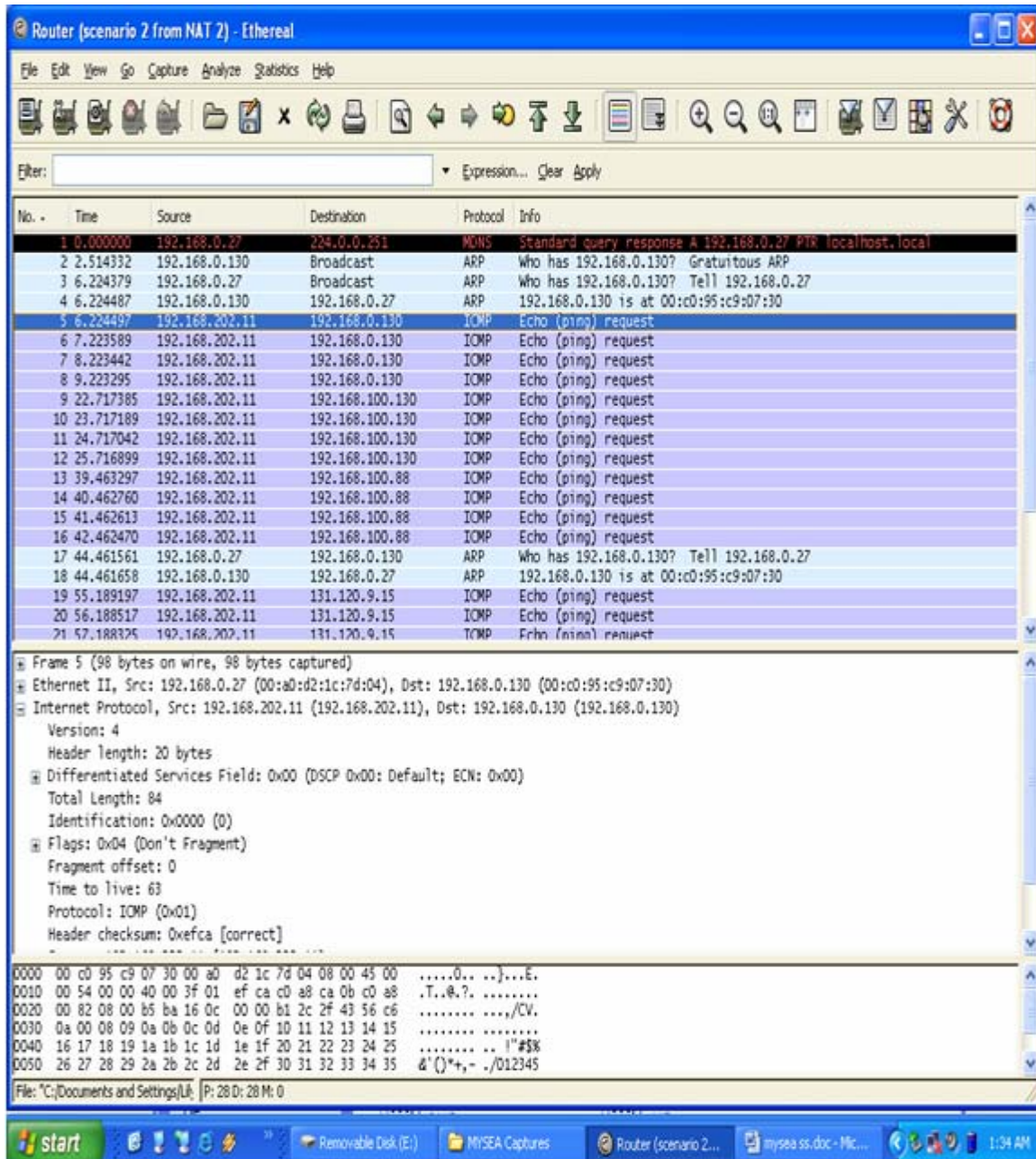


Figure 63. Test 6: Scenario 2 Packet Capture on Router (pinged from NAT 2)

E.6.2. NAT 1

The following four figures are snapshots of the packets captured on NAT 1 when the four interfaces were pinged from NAT 2.

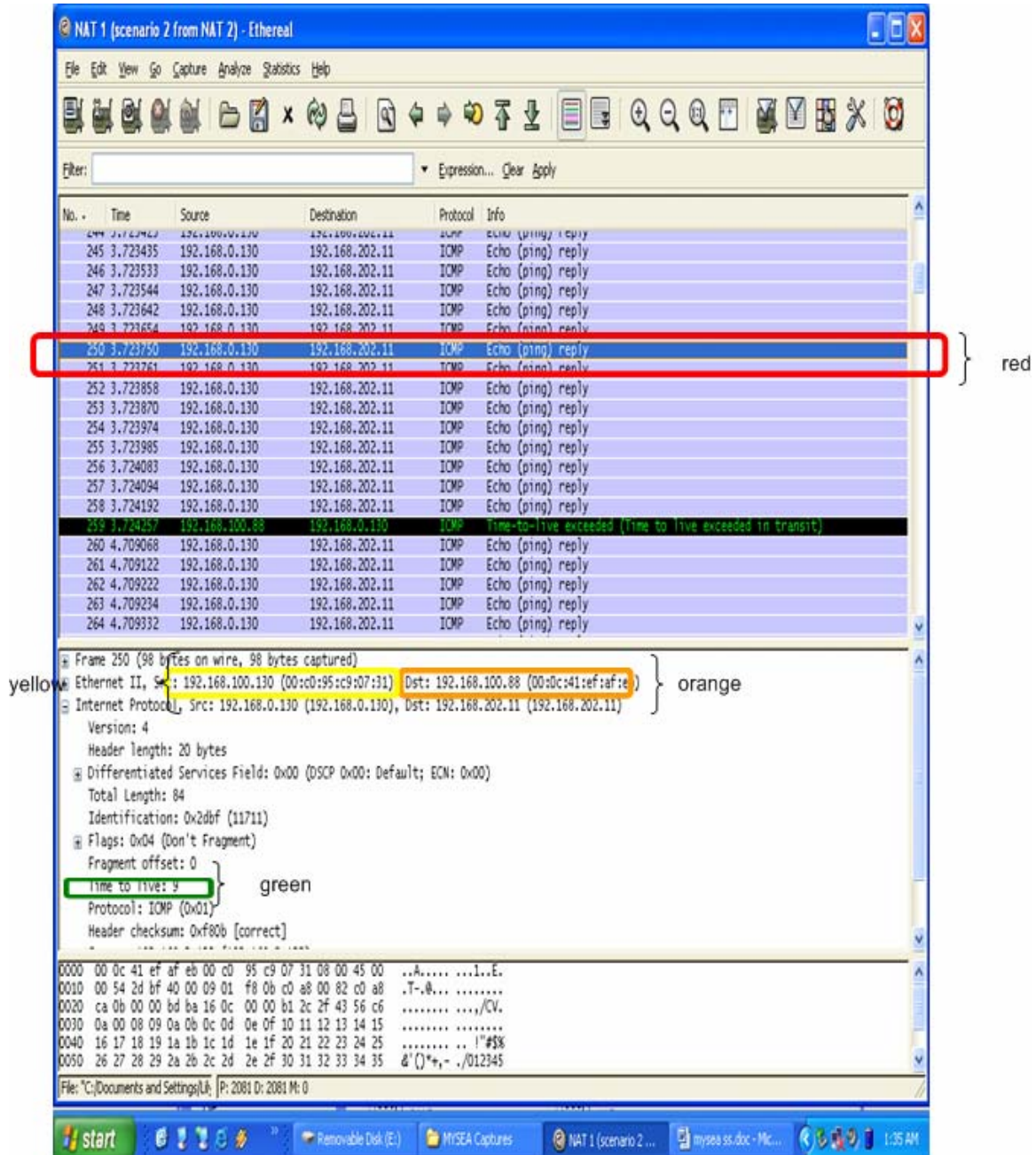


Figure 64. Test 6: Scenario 2 Packet Capture on NAT 1 (pinged from NAT 2), Part 1

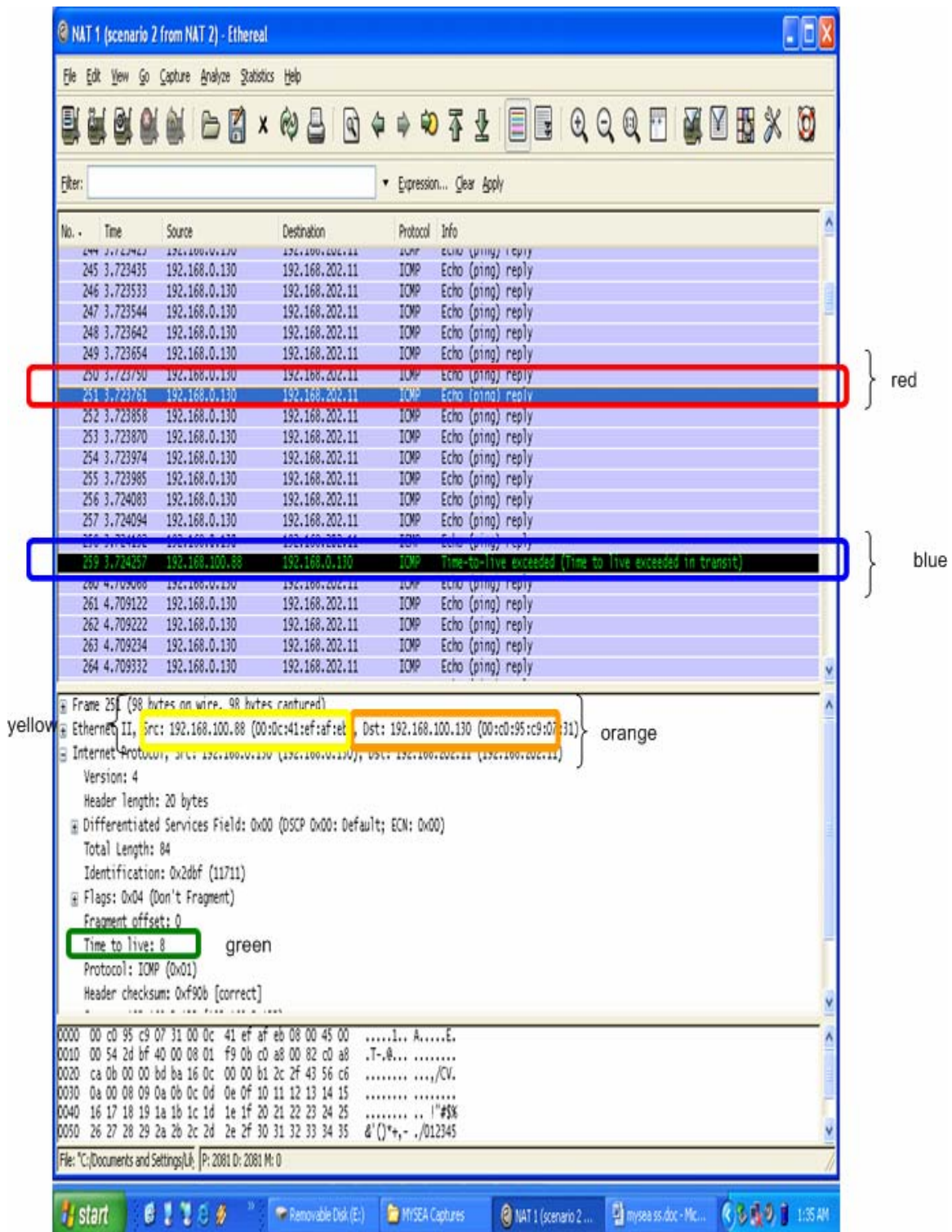


Figure 65. Test 6: Scenario 2 Packet Capture on NAT 1 (pinged from NAT 2), Part 2

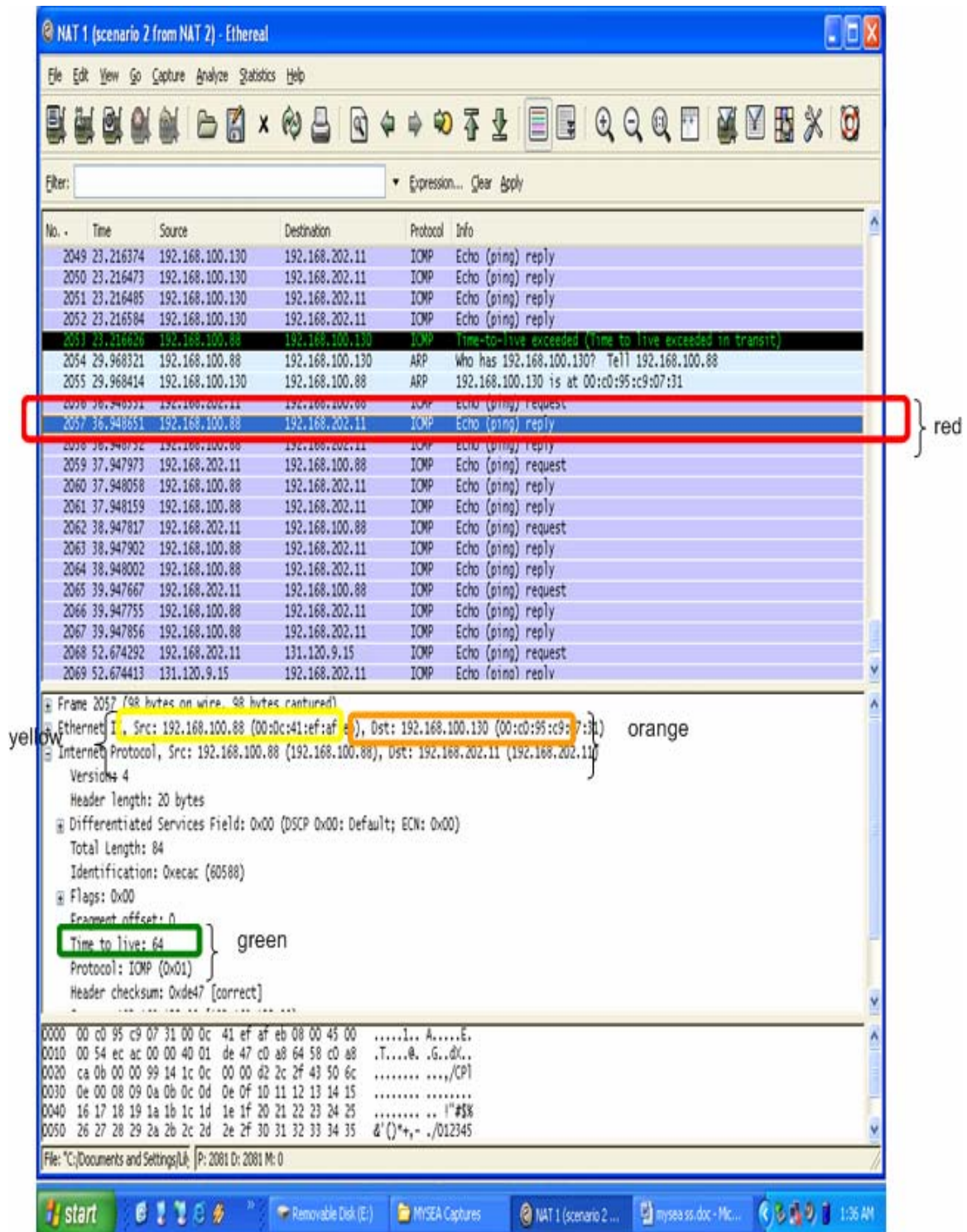


Figure 66. Test 6: Scenario 2 Packet Capture on NAT 1 (pinged from NAT 2), Part 3

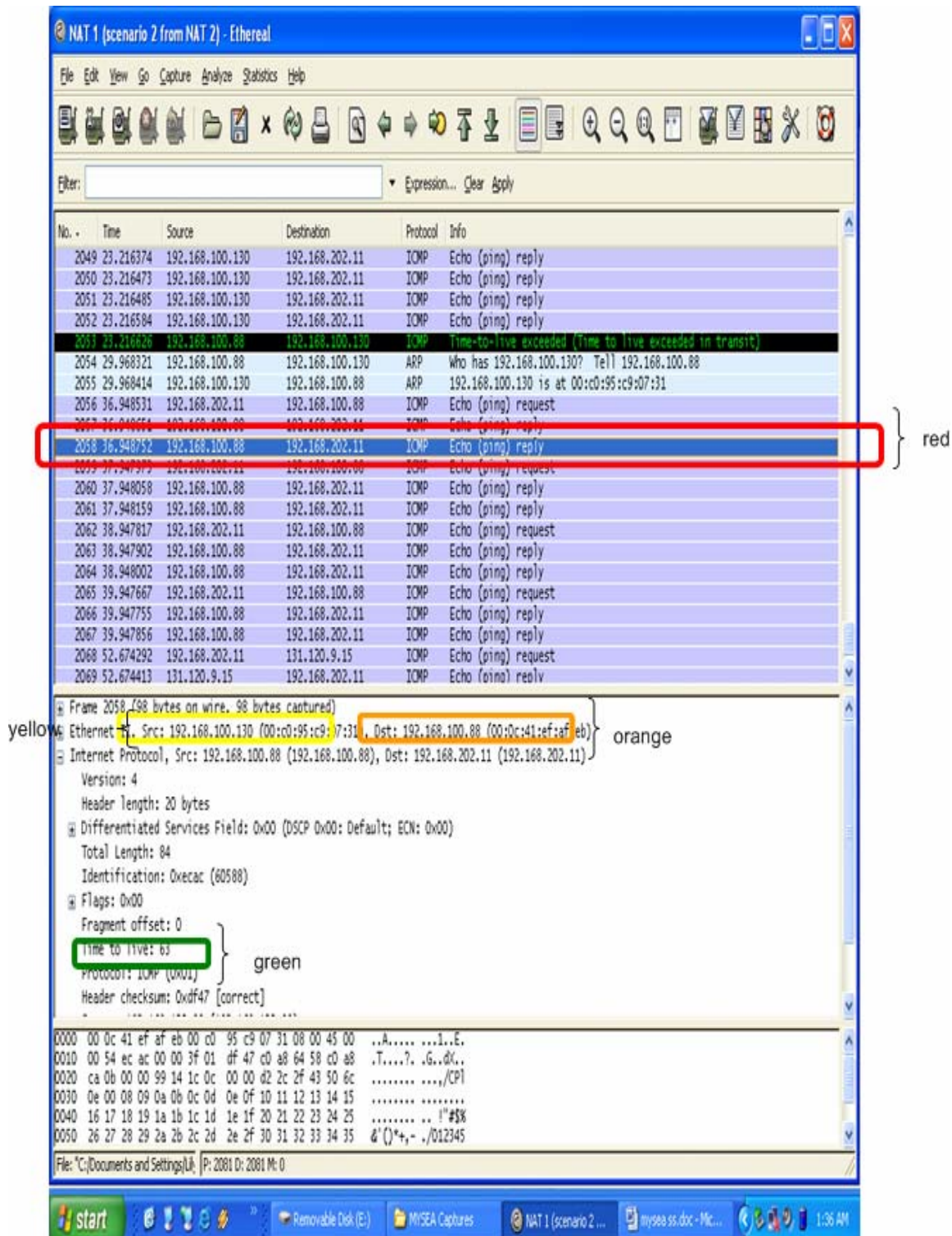


Figure 67. Test 6: Scenario 2 Packet Capture on NAT 1 (pinged from NAT 2), Part 4

E.6.3. Analysis

NAT 2 failed to ping 192.168.0.130 and 192.168.100.130.. Figure 63 show that the Router did not see any Echo replies from 192.168.0.130 or 192.168.100.130. However, it is evident that 192.168.0.130 and 192.168.100.130 replied since NAT 1 saw them (red outline in Figure 64). If routing was done correctly, the Echo replies should have been sent to the Router and then to NAT 2 instead of to NAT 1. Instead, MLS server sent the Echo replies to NAT 1. Figure 65 indicated that when NAT 1 received the Echo replies, it sent them to next hop (192.168.100.130) based on its routing table (yellow and orange outlines in Figure 65). However, XTS-400 forwarded the replies back NAT 1 at 192.168.100.88 (yellow and orange outlines in Figure 64). Hence, Echo replies were bounced back and forth between 192.168.100.130 and 192.168.100.88 until the time-to-live field reached zero (blue outline in Figure 65). This sequence of events also occurred for the Echo requests from 192.168.202.11 to 192.168.100.130.

NAT 2 also failed to ping 192.168.100.88 and 131.120.9.15. Figures 66 and 67 show that there exists two Echo replies for each Echo request destined for 192.168.100.88 and 131.120.9.15. For each Echo request destined for 192.168.100.88, NAT 1 responded with an Echo reply which is sent to its next hop at 192.168.100.130 (orange outline in Figure 66). However, the same Echo reply was bounced back by XTS-400 to where it came from when XTS-400 received it (yellow and orange outline in Figure 67). NAT 1 stopped routing the Echo reply further since it recognized its own packet. This explains why NAT 2 was never able to receive any replies. The same was true when NAT 2 pinged 131.120.9.15.

E.7. Packet Capture when pinged from Router

E.7.1. Router

The following is a snapshot of the packets captured on the Router when the four interfaces were pinged from the Router.

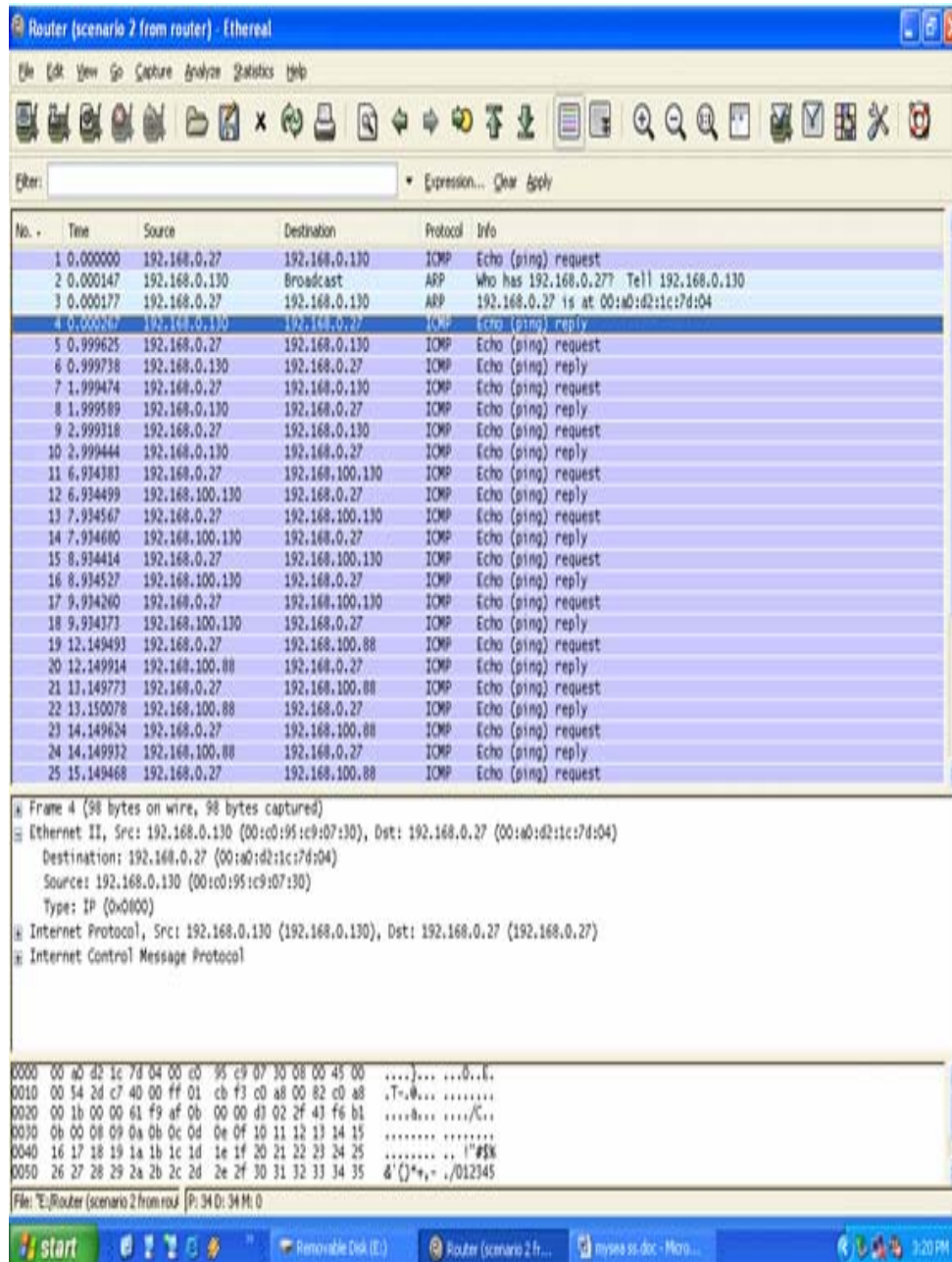


Figure 68. Test 6: Scenario 2 Packet Capture on Router (pinged from Router)

E.7.2. NAT 1

The following is a snapshot of the packets captured on NAT 1 when the four interfaces were pinged from the Router.

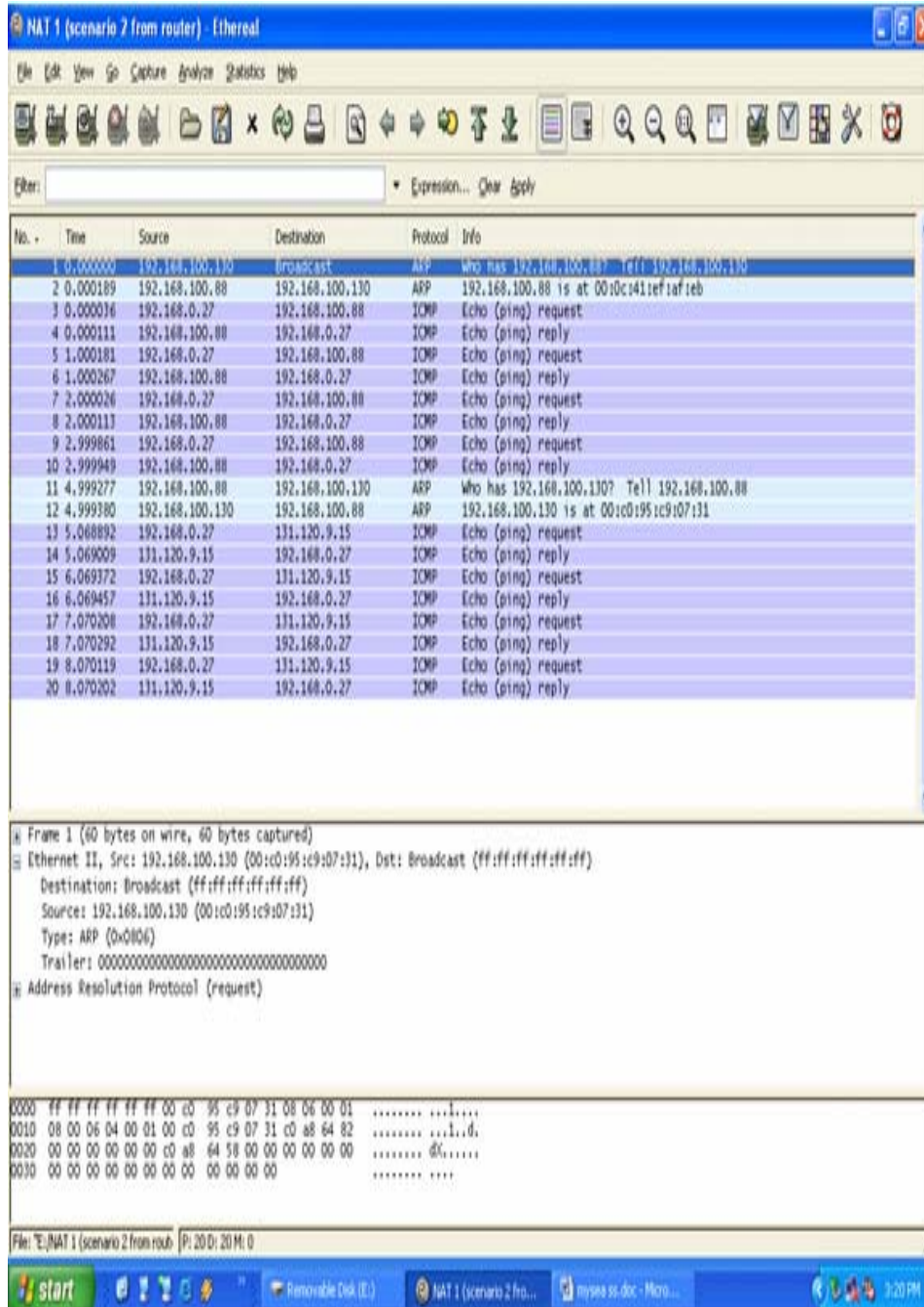


Figure 69. Test 6: Scenario 2 Packet Capture on NAT 1 (pinged from Router)

E.7.3. Analysis

The Router was able to ping all four interfaces described in D.2. The Router could ping 192.168.0.130 because it shares a peer-to-peer relationship with the MLS server. It was also able to ping 192.168.100.130 because XTS-400 knew its interfaces. The same logic applied to the case when the Router pinged 192.168.100.88. The Router successfully pinged 131.120.9.15 in this scenario but not in Scenario 1. In order for this to occur, XTS-400 would have to send the Echo requests to 192.168.100.88 in order for the requests to reach 131.120.9.15. When 131.120.9.15 received the requests, it sent Echo replies to the Router by routing the packet to the next hop or the MLS server. XTS-400 had the logic to route the replies to the Router since it knew its peers.

F. Scenario 3

F.1. Description

The MLS server is configured in order as follows: any packets received on its single-level interface (eth1, 192.168.100.130) is forwarded to the single-level interface of NAT 1 (192.168.100.88) and any packets received on its MLS LAN interface (192.168.0.130) is forwarded to the public interface of the Router (192.168.0.27).

F.2. Operations

First, NAT 2 pings:

1. eth0 of MLS server
2. eth1 of MLS server
3. eth1 of NAT 1
4. eth0 of NAT 1

Then, Router pings:

5. eth0 of MLS server
6. eth1 of MLS server
7. eth1 of NAT 1
8. eth0 of NAT 1

F.3. Network Configuration on MLS Server

F.3.1. Type the following answers when prompted:

SAK

Enter command?	tcPIP_edit
Enter editor request?	add
Enter TCP/IP daemon name?	tcPIP_mls
Enter TCPIP/IP daemon description?	TCP/IP for MLS LAN network
Enter domain name?	cisrlabmlstestbed1.com
Enter host name?	mlsServer
Enable the subnets local flag?	n
Enable the IP forwarding flag?	y
Enable the IP send redirect flag?	y
Enable the shutdown on failure flag?	n
Use default TCP maximum retransmission?	y
Add the network interface configuration?	y
Enter TCP/IP device name?	/dev/ether0
Enter interface address?	192.168.0.130
Enter destination address?	0.0.0.0
Enter broadcast address?	192.168.0.255
Enter network mask?	255.255.255.0
Add another network interface entry?	y
Enter TCP/IP device name?	/dev/ether1
Enter interface address?	192.168.100.130
Enter destination address?	0.0.0.0
Enter broadcast address?	192.168.100.255
Enter network mask?	255.255.255.0
Add another network interface entry?	n
Add the route configuration?	y
Enter TCP/IP device name	/dev/ether1
Is this route a default route	n
Enter destination address	0.0.0.0
Is destination address a host	n

Enter gateway address	192.168.100.88
Enter route metric	1
Add another network route entry	y
Enter TCP/IP device name	/dev/ether0
Is this route a default route	n
Enter destination address	0.0.0.0
Is destination address a host	n
Enter gateway address	192.168.0.27
Enter route metric	1
Add another network route entry	n
Add the resolver configuration?	n

F.4. Preparation and Testing

F.4.1. On NAT 1,

- F.4.1.1. Launch Ethereal
- F.4.1.2. Go to the **Capture** menu
- F.4.1.3. Go to **Interfaces**
- F.4.1.4. Click on **Capture 192.168.100.88**

F.4.2. On Router,

- F.4.2.1. Launch Ethereal
- F.4.2.2. Go to the **Capture** menu
- F.4.2.3. Go to **Interfaces**
- F.4.2.4. Click on **Capture 192.168.0.27**

F.4.3. On NAT 2,

F.4.3.1. Run the following commands:

```
ping -c 4 192.168.0.130
ping -c 4 192.168.100.130
ping -c 4 192.168.100.88
ping -c 4 131.120.9.15
```

F.4.4. Repeat the above commands on the Router

F.4.5. Stop Ethereal captures on both NAT 1 and Router

F.5. Result

Table 13 lists the result of the Scenario 3. The first column shows where the ping was initiated and the first row shows what hosts/IP addresses were pinged. The result is exactly the same as the result obtained in Scenario 2.

to from	192.168.0.130 (eth0, MLS LAN interface of MLS server)	192.168.100.130 (eth1, single-level interface of MLS server)	192.168.100.88 (eth1, single-level interface of NAT 1)	131.120.9.15 (eth0, public interface of NAT 1)
NAT 2	Failed	Failed	Failed	Failed
Router	Successful	Successful	Successful	Successful

Table 13. Test 6: Scenario 3 Result

F.6. Packet Capture when pinged from NAT 2

F.6.1. Router

The following is a snapshot of the packets captured on the Router when the four interfaces were pinged from NAT 2.

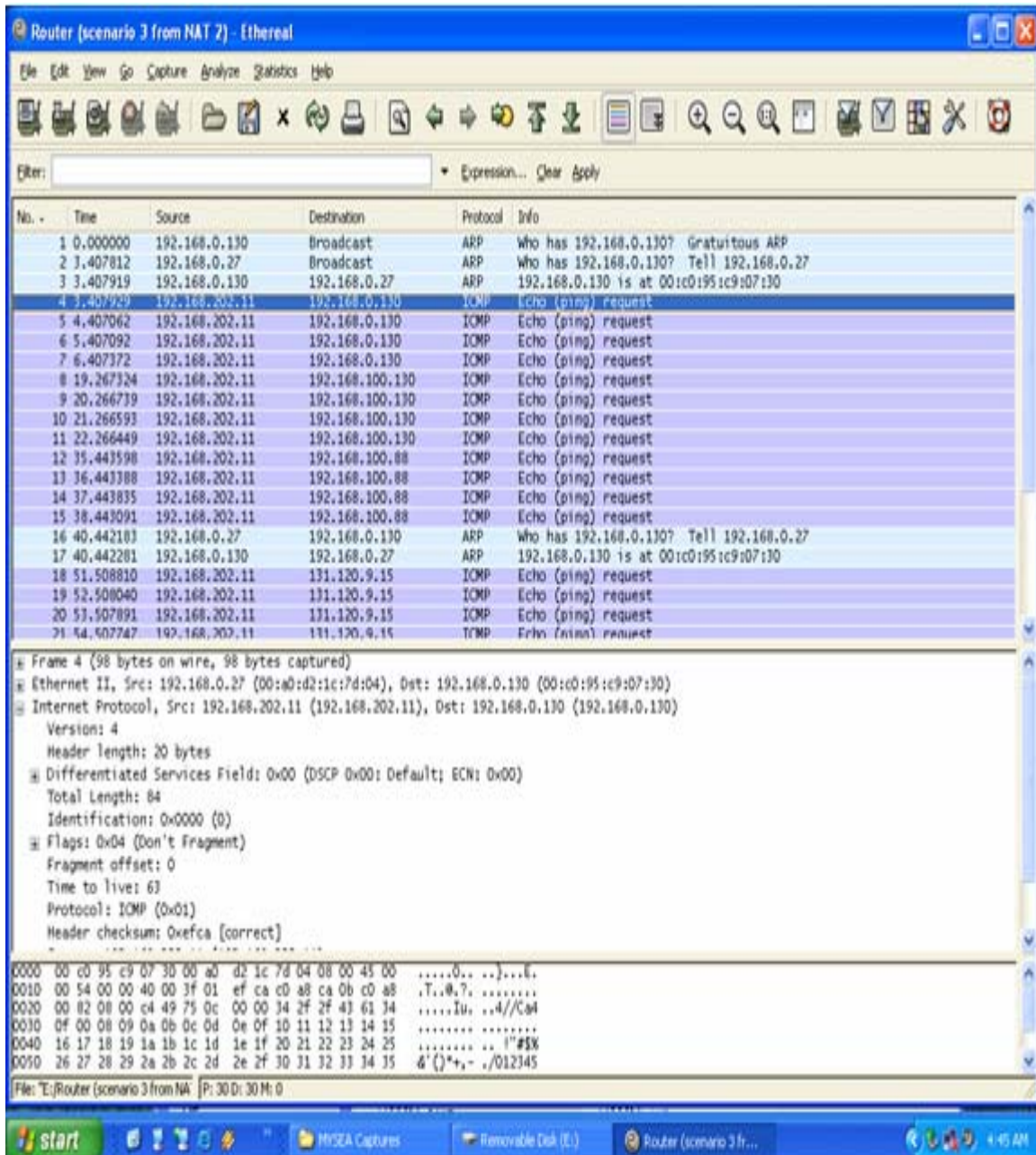


Figure 70. Test 6: Scenario 3 Packet Capture on Router (pinged from NAT 2)

F.6.2. NAT 1

The following three figures are snapshots of the packets captured on NAT 1 when the four interfaces were pinged from NAT 2.

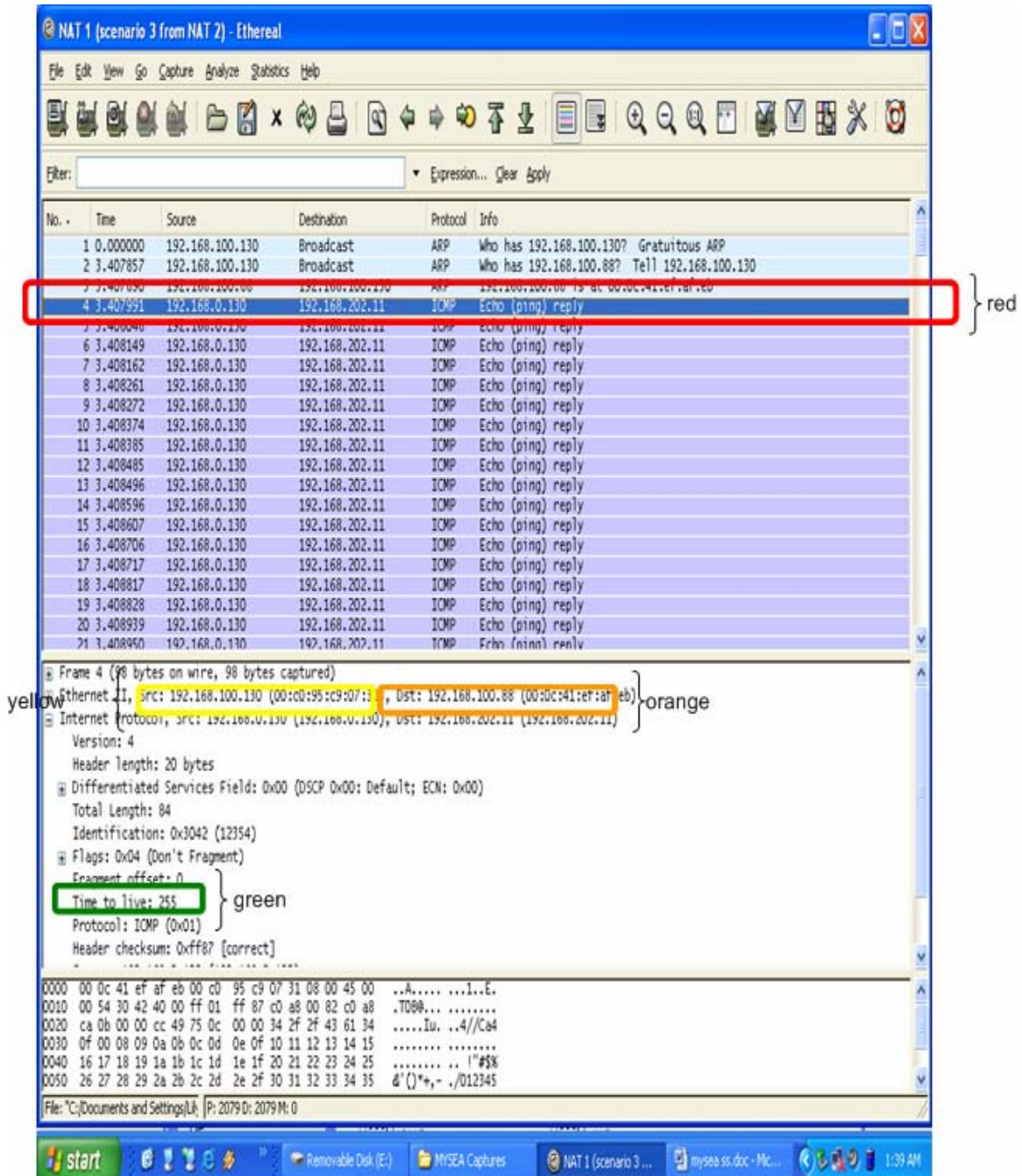
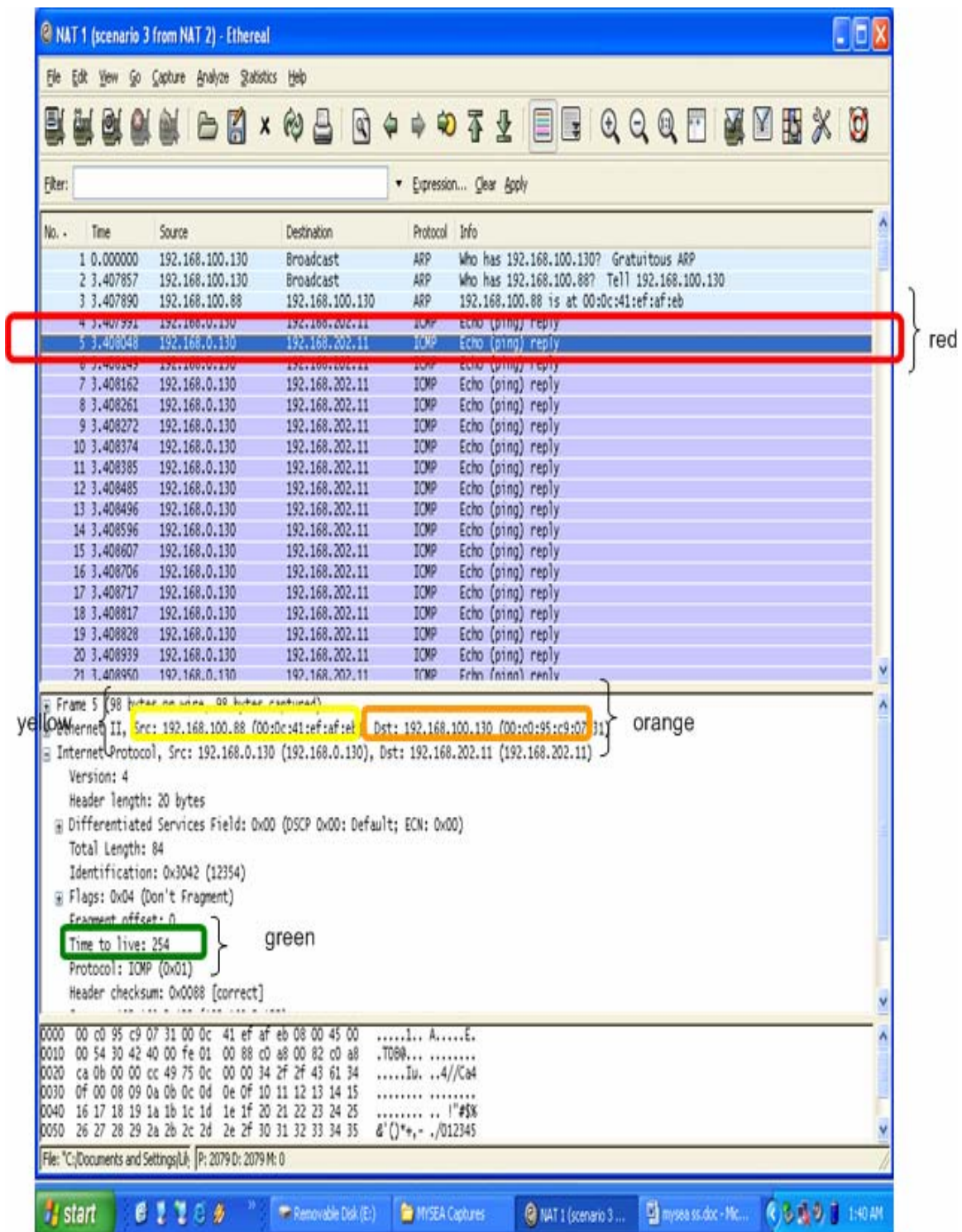


Figure 71. Test 6: Scenario 3 Packet Capture on NAT 1 (pinged from NAT 2), Part 1



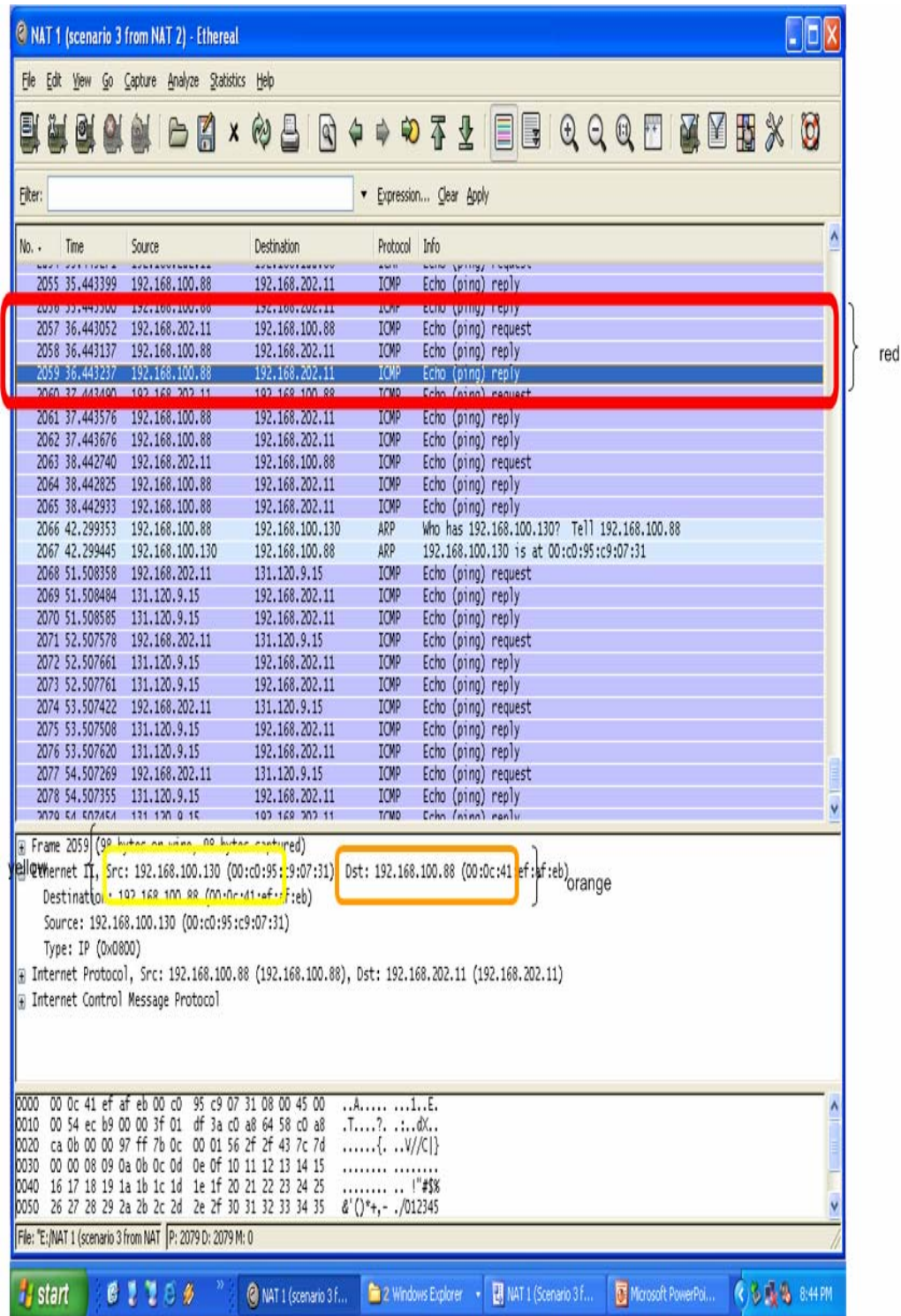


Figure 73. Test 6: Scenario 3 Packet Capture on NAT 1 (pinged from NAT 2), Part 3

F.6.3. Analysis

NAT 2 failed to ping all four interfaces. The behaviors seen in the packet captures in this section are identical to the behaviors seen in section E.5. All the Echo replies (red outline in Figure 71) from 192.168.0.130 and 192.168.100.130 were bounced between 192.168.100.130 and 192.168.100.88 (yellow and orange outlines in Figure 71 and Figure 72). As a result, the replies never reached the Router (Figure 70). As described in F.5, every Echo request destined for 192.168.100.88 and 131.120.9.15 had two Echo replies (red outline in Figure 72). The first reply went from 192.168.100.88 to 192.168.100.130. When the MLS server received the reply, the XTS-400 routed it back to 192.168.100.88. This was the reason for seeing two Echo replies per request. Refer to E.6.3 for more detail explanation.

F.7. Packet Capture when pinged from Router

F.7.1. Router

The following is a snapshot of the packets captured on the Router when the four interfaces were pinged from Router.

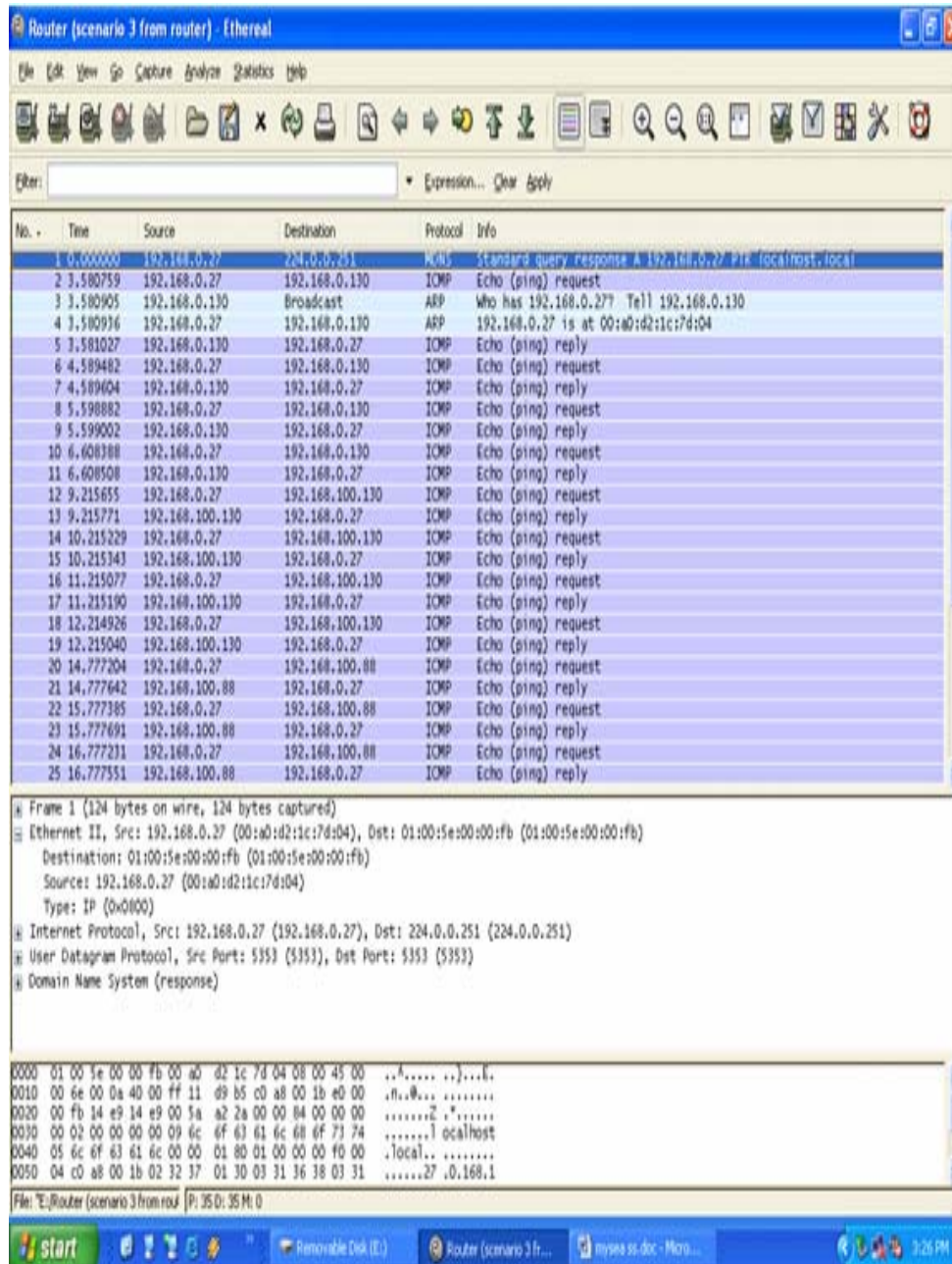


Figure 74. Test 6: Scenario 3 Packet Capture on Router (pinged from Router)

F.7.2. NAT 1

The following is a snapshot of the packets captured on NAT 1 when the four interfaces were pinged from Router.

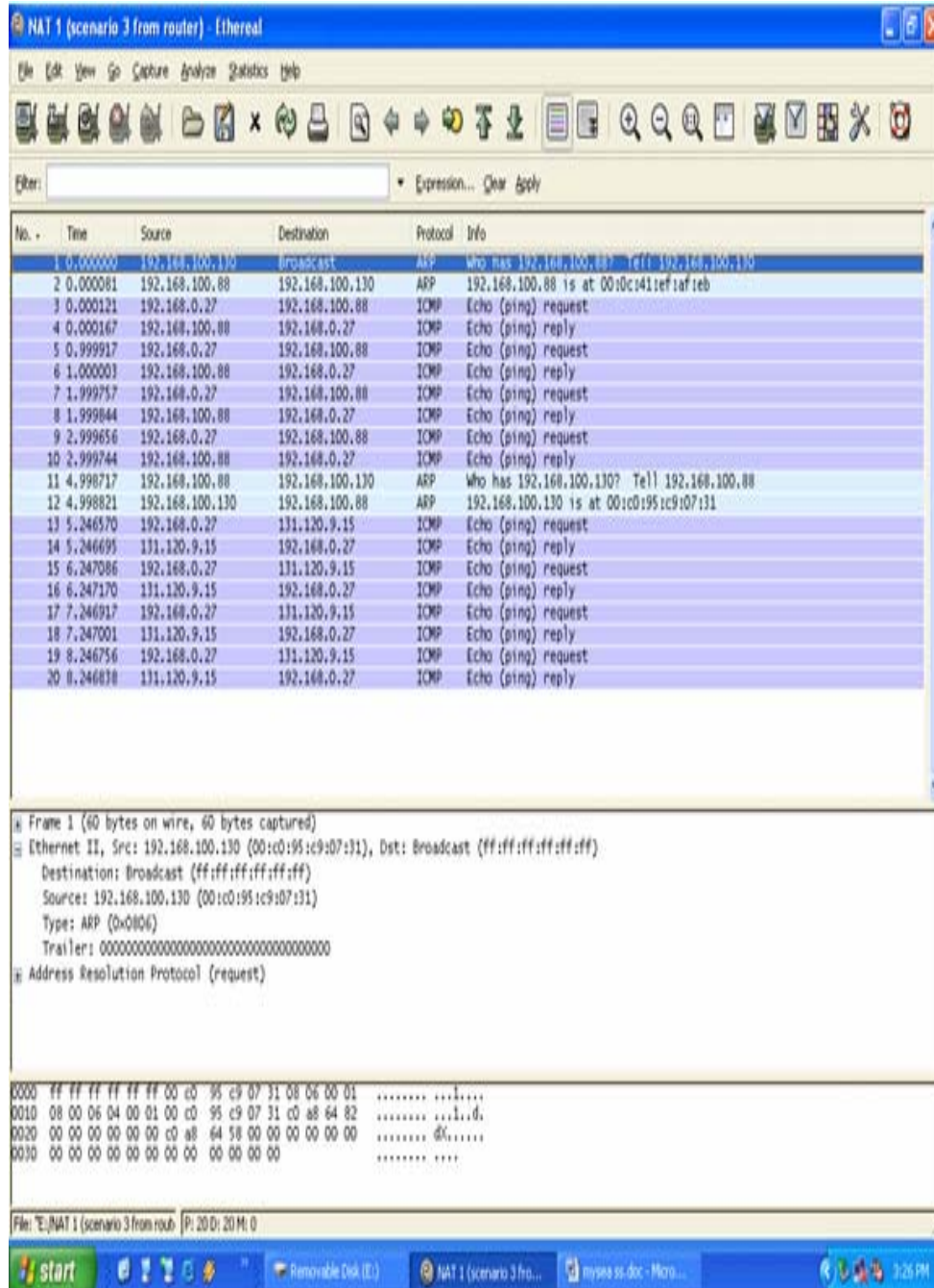


Figure 75. Test 6: Scenario 3 Packet Capture on NAT 1 (pinged from Router)

F.7.3. Analysis

The Router was able to ping all four interfaces as described in F.2. The behavior of this scenario is identical to the one in E.7. Refer to E.7.3 for a more detail analysis.

G. Scenario 4

G.1. Description

The MLS server is configured as follows in order: any packets received on its single-level interface (eth1, 192.168.100.130) is forwarded to the single-level interface of the Router (eth0, 192.168.0.27) and any packets received on its MLS LAN interface (eth0, 192.168.0.130) is forwarded to the single-level interface of the NAT 1 (eth1, 192.168.100.88).

G.2. Operations

First, NAT 2 pings:

1. eth0 of MLS server
2. eth1 of MLS server
3. eth1 of NAT 1
4. eth0 of NAT 1

Then, Router pings:

5. eth0 of MLS server
6. eth1 of MLS server
7. eth1 of NAT 1
8. eth0 of NAT 1

G.3. Network Configuration on MLS Server.

G.3.1. Type the following answers when prompted:

SAK

Enter command?	tcpip_edit
Enter editor request?	add
Enter TCP/IP daemon name?	tcpip_mls
Enter TCPIP/IP daemon description?	TCP/IP for MLS LAN
network	
Enter domain name?	cisrlabmlstestbed1.com

Enter host name?	mlsserver
Enable the subnets local flag?	n
Enable the IP forwarding flag?	y
Enable the IP send redirect flag?	y
Enable the shutdown on failure flag?	n
Use default TCP maximum retransmission?	y
Add the network interface configuration?	y
Enter TCP/IP device name?	/dev/ether0
Enter interface address?	192.168.0.130
Enter destination address?	0.0.0.0
Enter broadcast address?	192.168.0.255
Enter network mask?	255.255.255.0
Add another network interface entry?	y
Enter TCP/IP device name?	/dev/ether1
Enter interface address?	192.168.100.130
Enter destination address?	0.0.0.0
Enter broadcast address?	192.168.100.255
Enter network mask?	255.255.255.0
Add another network interface entry?	n
Add the route configuration?	y
Enter TCP/IP device name	/dev/ether1
Is this route a default route	n
Enter destination address	0.0.0.0
Is destination address a host	n
Enter gateway address	192.168.0.27
Enter route metric	1
Add another network route entry	y
Enter TCP/IP device name	/dev/ether0
Is this route a default route	n
Enter destination address	0.0.0.0
Is destination address a host	n

Enter gateway address	192.168.100.88
Enter route metric	1
Add another network route entry	n
Add the resolver configuration?	n

G.4.Preparation and Testing

G.4.1. On NAT 1,

- G.4.1.1. Launch Ethereal
- G.4.1.2. Go to the **Capture** menu
- G.4.1.3. Go to **Interfaces**
- G.4.1.4. Click on **Capture 192.168.100.88**

G.4.2. On Router,

- G.4.2.1. Launch Ethereal
- G.4.2.2. Go to the **Capture** menu
- G.4.2.3. Go to **Interfaces**
- G.4.2.4. Click on **Capture 192.168.0.27**

G.4.3. On NAT 2,

G.4.3.1. Run the following commands:

```
ping -c 4 192.168.0.130
ping -c 4 192.168.100.130
ping -c 4 192.168.100.88
ping -c 4 131.120.9.15
```

G.4.4. Repeat the above commands on the Router

G.4.5. Stop Ethereal captures on both NAT 1 and Router

G.5.Result

Table 14 lists the result of the Scenario 4. The first column shows where the ping was initiated and the first row shows what hosts/IP addresses were pinged. The result from Scenario 4 is exactly the same as the result obtained in Scenario 1.

to from	192.168.0.130 (eth0, MLS LAN interface of MLS server)	192.168.100.130 (eth1, single-level interface of MLS server)	192.168.100.88 (eth1, single-level interface of NAT 1)	131.120.9.15 (eth0, public interface of NAT 1)
NAT 2	Successful	Successful	Successful	Failed
Router	Successful	Successful	Successful	Failed

Table 14. Test 6: Scenario 4 Result

G.6. Packet Capture when pinged from NAT 2

G.6.1. Router

The following is a snapshot of the packets captured on the Router when the four interfaces were pinged from NAT 2.

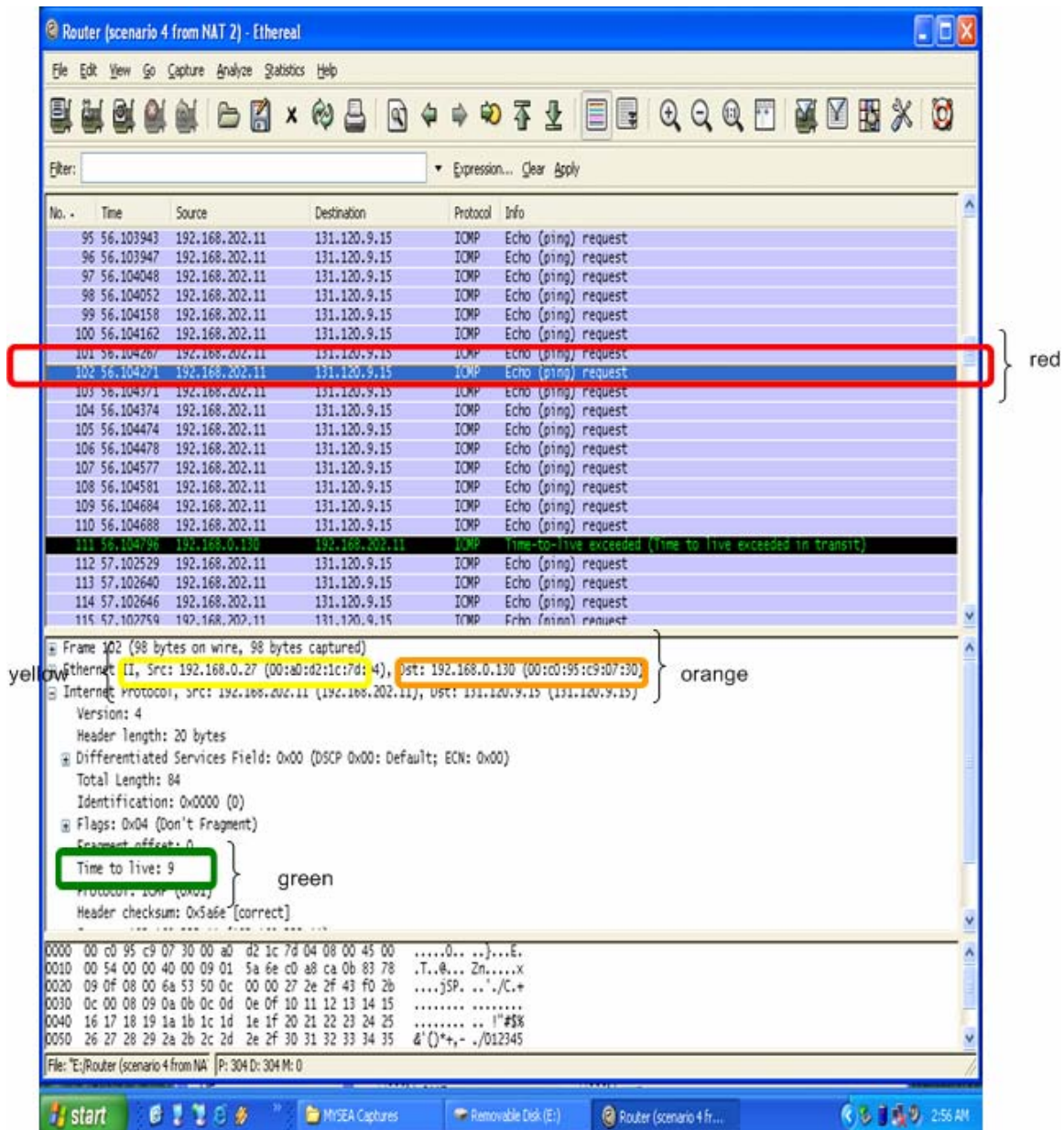


Figure 76. Test 6: Scenario 4 Packet Capture on Router (pinged from NAT 2), Part 1

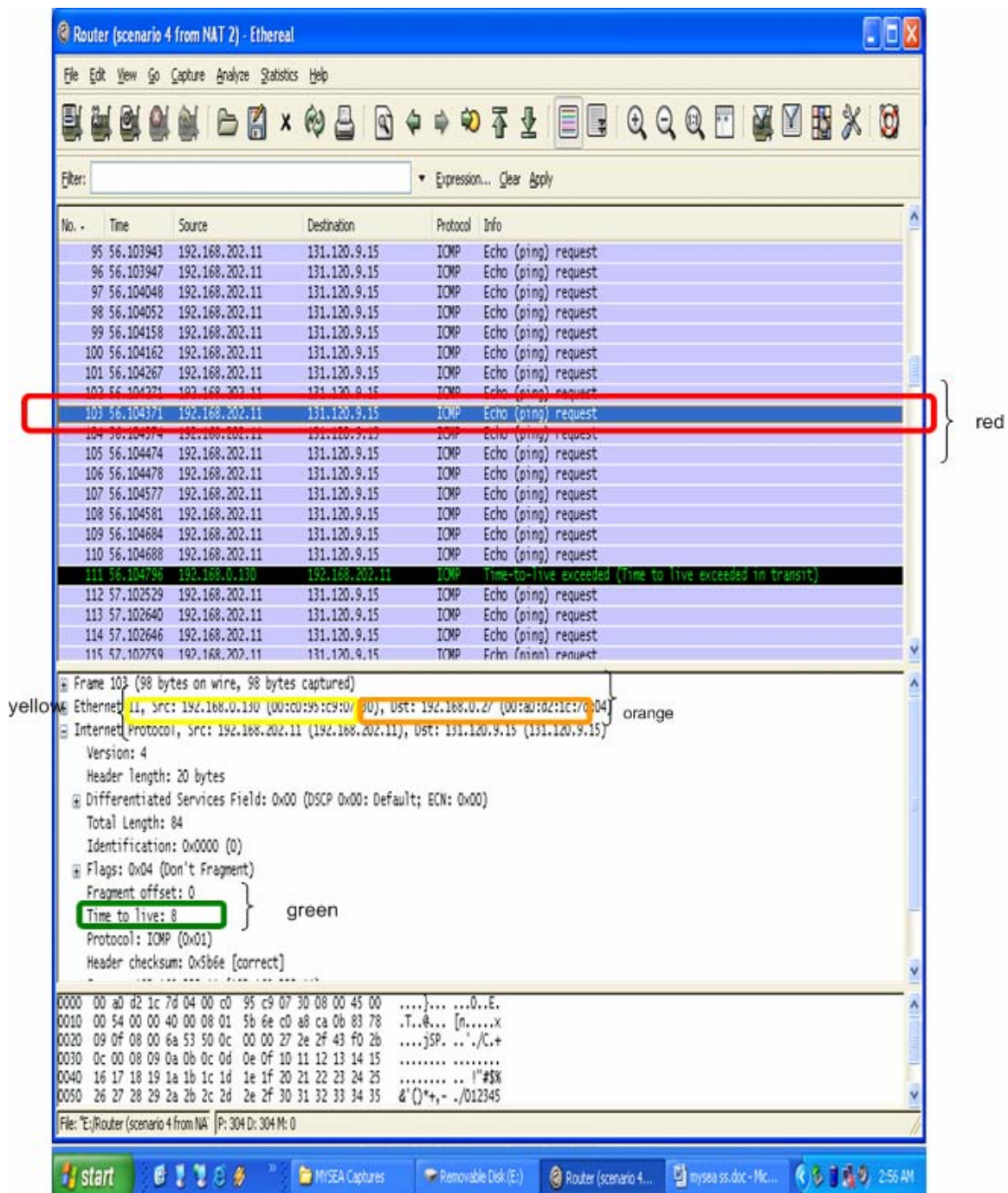


Figure 77. Test 6: Scenario 4 Packet Capture on Router (pinged from NAT 2), Part 2

G.6.2. NAT 1

The following is a snapshot of the packets captured on the NAT 1 when the four interfaces were pinged from NAT 2.

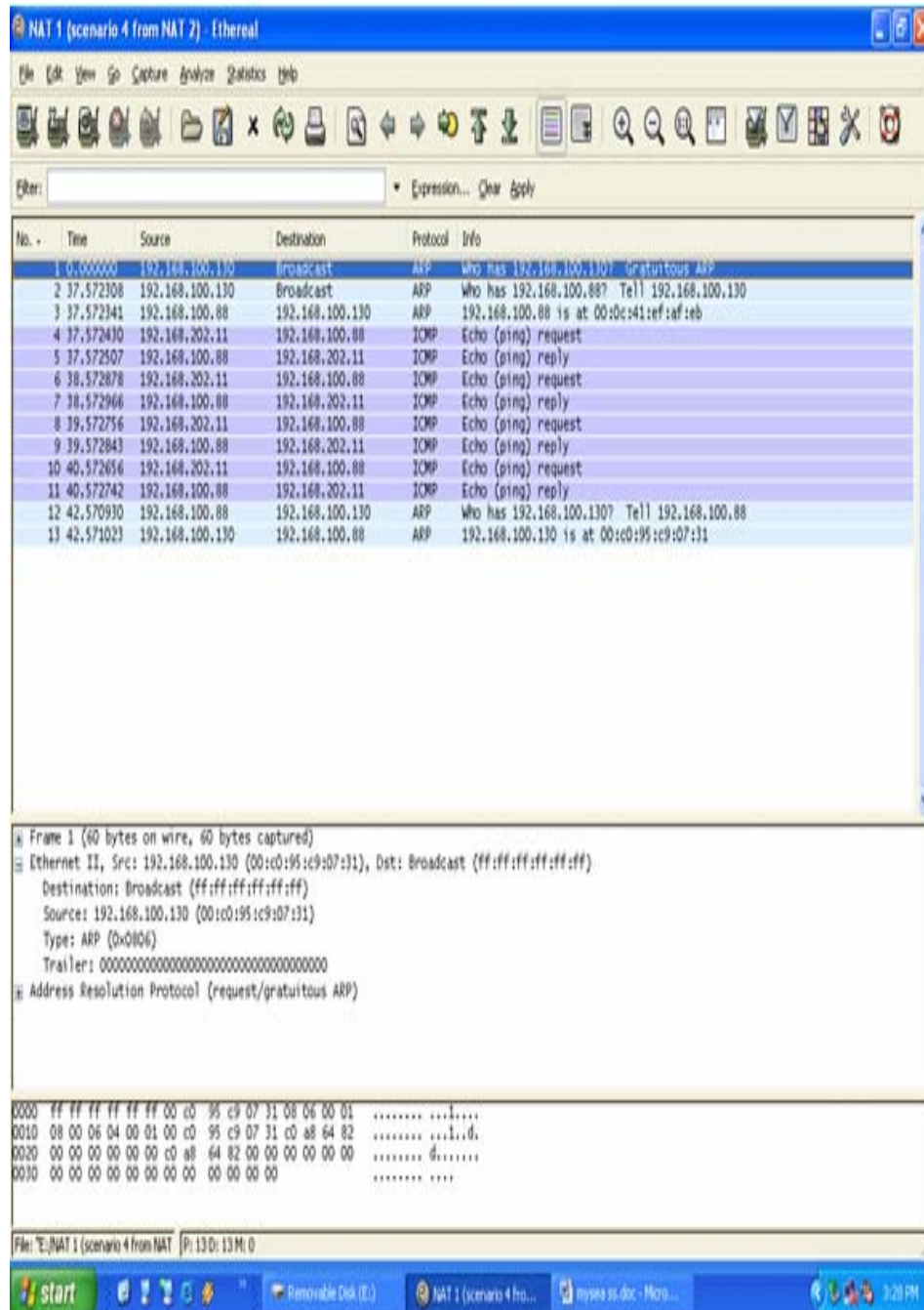


Figure 78. Test 6: Scenario 4 Packet Capture on NAT 1 (pinged from NAT 2)

G.6.3. Analysis

Scenario 4 had the same result as Scenario 1. In other words, NAT 2 was able to ping 192.168.0.130, 192.168.100.130, and 192.168.100.88 only. It failed to ping 131.120.9.15. The Echo request destined for 131.120.9.15 was routed from 192.168.0.27 to 192.168.0.130 (yellow and orange outlines in Figure 76). However, XTS-400 forwarded the packet back to 192.168.0.27. This sequence of events occurred until the time-to-live exceeded. As a result, the Echo requests never reached NAT 1 (Figure 77). Refer to D.6.3 for more details.

G.7. Packet Capture when pinged from Router

G.7.1. Router

The following is a snapshot of the packets captured on the Router when the four interfaces were pinged from Router.

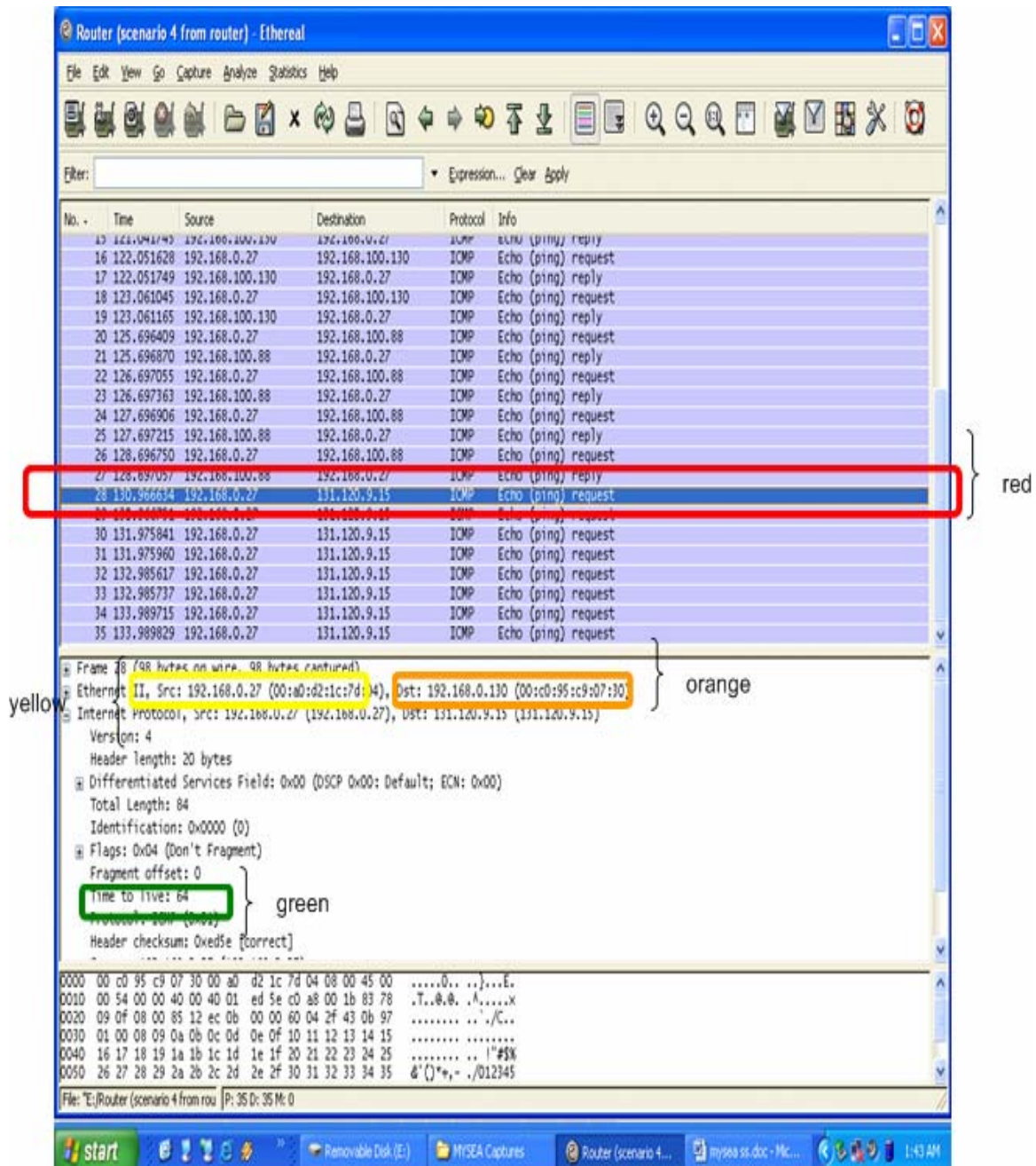


Figure 79. Test 6: Scenario 4 Packet Capture on Router (pinged from Router), Part 1

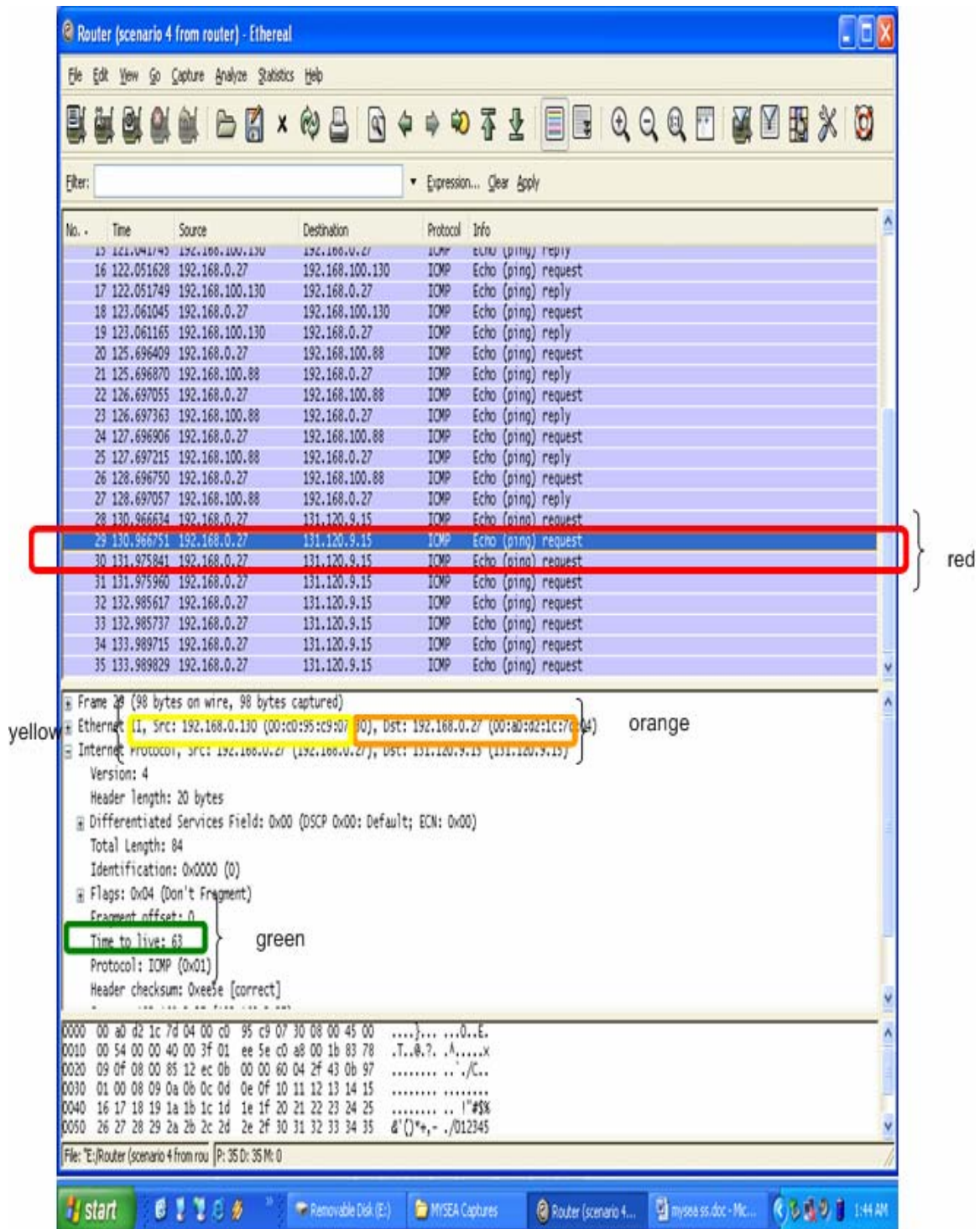


Figure 80. Test 6: Scenario 4 Packet Capture on Router (pinged from Router), Part 2

G.7.2. Router

The following is a snapshot of the packets captured on the NAT 1 when the four interfaces were pinged from Router.

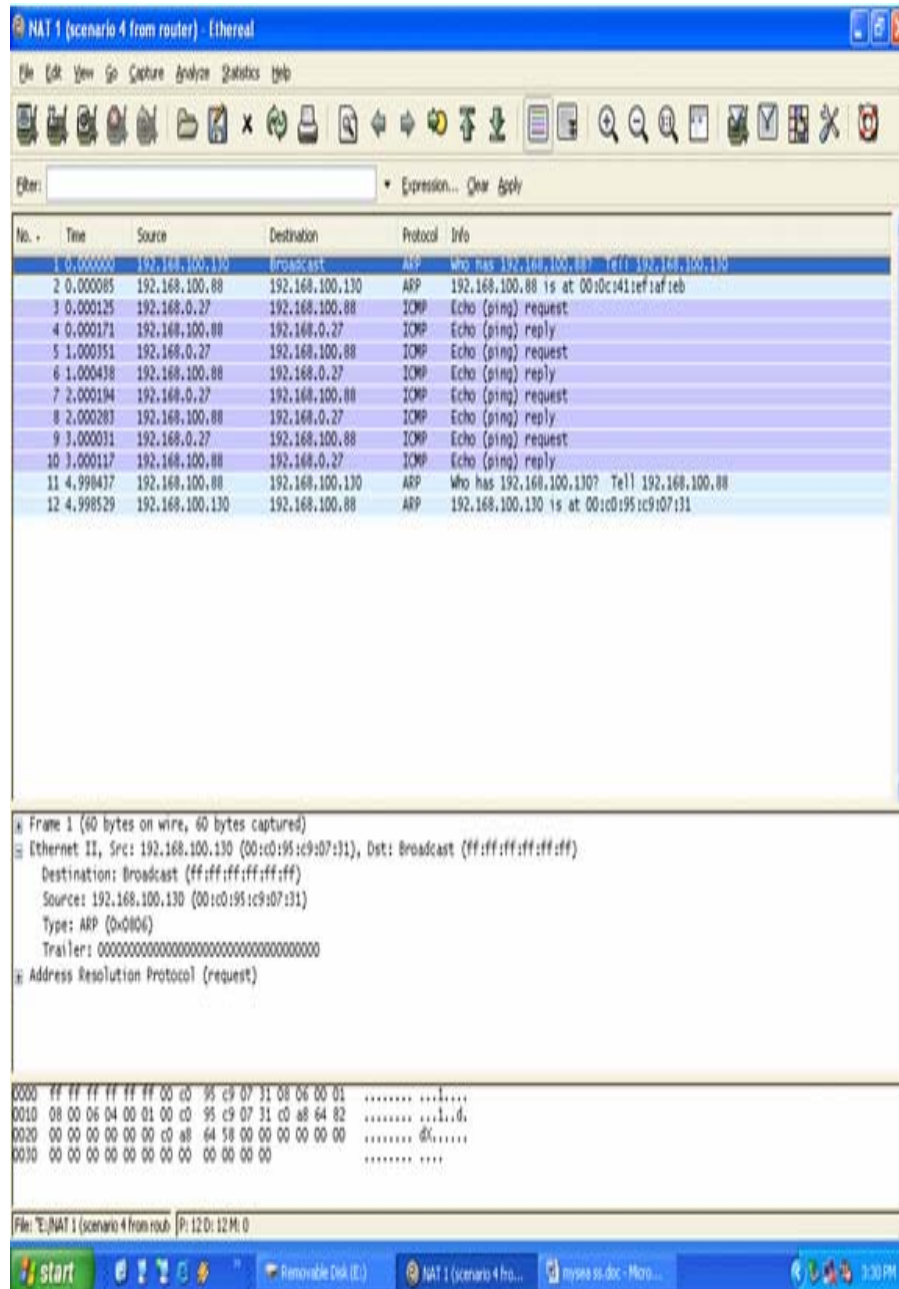


Figure 81. Test 6: Scenario 4 Packet Capture on NAT 1 (pinged from Router)

G.7.3. Analysis

The Router could ping both interfaces of the MLS server because the Router and the MLS server share a peer-to-peer relationship. The Router was also able to ping NAT 1 since XTS-400 has routing capabilities to route packets to its immediate peers. However, pinging 131.120.9.15 was unsuccessful. Echo requests destined (red outline in Figure 79) for 131.129.9.15 were sent out by the Router (yellow and orange outlines in Figure 79). However, XTS-400 sent those requests back to the Router when it received them (yellow and orange outlines in Figure 80). The Router stopped routing the requests further as it recognized them. Therefore, the Echo requests never reached NAT 1 (Figure 81).

H. Observation

A number of observations can be made from analyzing the results and packet captures for the four scenarios. First, there were two different sets of results from running the four test scenarios. Scenarios 1 and 4 generated the first set and scenarios 2 and 3 have generated the other set of results (refer to sections D.5, E.5, F.5, and G.5). Second, each scenario that shared the same gateway address sequence in its routing configuration yielded identical results. In other words, the results were dependent on the order of the gateway address and were independent of the device name in the routing configuration (refer to Table 9). Third, XTS-400 seemed to always forward packets destined for unknown networks to the gateway indicated in the first static route in its routing table. In scenarios 1 and 4, XTS-400 bounced Echo requests it received from 131.120.9.15 to 192.168.0.27 instead of forwarding them onto NAT 1. Also in scenarios 1 and 4, similar behavior of XTS-400 forwarding packets to 192.168.0.27 was seen when the Router pinged 131.120.9.15. In both cases, the XTS-400 did not have routing information for the 131.120.9.x network and the gateway for its first static route is 192.168.0.27. The XTS-400 always routed packets destined for unknown networks to 192.168.100.88 instead in scenarios 2 and 3 where the 192.168.100.88 was the gateway in its first static route.

LIST OF REFERENCES

- [1] T. Richardson, "US to embrace VoIP", 2005.
http://www.theregister.co.uk/2005/04/04/idc_voip_research/, Accessed: April 2005.
- [2] ZDNet Research. "Corporate VOIP spending to reach \$903 mln in 2005" 2005
Available: <http://blogs.zdnet.com/ITFacts/index.php?id=P2656>, Accessed: April 2005.
- [3] VOIP-info.org, "VOIP Phones", 2005, <http://www.voip-info.org/wiki-VOIP+Phones#id892699>, Accessed: April 2005.
- [4] P. C. Mehta and S. Udani, "Overview of Voice over IP", Technical Report MS-CIS-01-31, Department of Computer Information Science, University of Pennsylvania, February 2001.
- [5] D. R. Kuhn, T. J. Walsh, and S. Fires, "Security Consideration for Voice over IP Systems, Recommendations of the National Institute of Standards and Technology", Special Publication 800-85, 2005.
- [6] "netfilter", 2005, <http://www.netfilter.org/>, Accessed: August 2005.
- [7] C. E. Irvine, T. E. Levin, T. D. Nguyen, D. Shifflett, J. Khosalim, P. C. Clark, A. Wong, F. Afinidad, D. Bibighaus, J. Sears, "Overview of a High Assurance Architecture for Distributed Multilevel Security," Proceedings of the 5th IEEE Systems, Man and Cybernetics Information Assurance Workshop, United States Military Academy, West Point, NY, pg 38-45, June 10-11, 2004.
- [8] T. D. Nguyen, T. E. Levin, and C. E. Irvine, "MYSEA Testbed", Proceedings from the 6th IEEE Systems, Man and Cybernetics Information Assurance Workshop, West Point, NY, June 2005, pp. 438-439.
- [9] J. Glasmann, W. Kellerer, and H. M"uller, "Service Architectures in H.323 and SIP: A Comparison", IEEE Communications Society Survey and Tutorials, Vol. 5, No. 2, 2003.

- [10] H. Schulzrinne and J. Rosenberg, "A Comparison of SIP and H.323 for Internet Telephony", Proceedings of the 8th International Workshop on Network and Operating System Support for Digit Audio and Video (NOSSDAV 98), Cambridge, England, Jul. 1998, pp. 83-86.
- [11] Javvin Technologies, Inc., "H.323: ITU-T VOIP Protocols Overview", 2005, <http://www.javvin.com/protocolH323.html>, Accessed: June 2005.
- [12] N. Networks, "A Comparison of H.323v4 and SIP", 3GPP S2, Tokyo, Japan, Technical Report S2-000505, January 2000.
- [13] S. Niccolini, R. G. Garroppo, J. Ott, S. Prella, J. Kuthan, S. Ubik, M. Brandl, D. Daskopoulos, E. Verharen, E. Dobbelssteijn, "IP Telephony Cookbook", pp. 76, TERENA, 2004. Available at: <http://www.terena.nl/library/IPTELEPHONYCOOKBOOK/chapters/IPTELEPHONYCOOKBOOK.pdf>, Accessed August 2005.
- [14] J. Winters, "IP Videoconferencing: ITU's H.323 and IETF's SIP", 2003 <http://www.eng.mu.edu/rehab/Rehab167/Mod3/teleconf/h323-sip.htm>, Accessed: April 2005.
- [15] "SJ Labs", 2005, <http://www.sjlabs.com>, Accessed: August 2005.
- [16] "Ethereal", 2005, <http://www.ethereal.com>, Accessed: August 2005.
- [17] "Zone Labs" 2005 <http://www.zonelabs.com/store/content/home.jsp>, Accessed August 2005.
- [18] John G. Ata, private email, 16 September 2005.
- [19] J. Stephens, "Connection tracking", 2001, <http://kalamazoolinux.org/presentations/20010417/contrack.html>, Accessed: September 2005.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, VA
2. Dudley Knox Library
Naval Postgraduate School
Monterey, CA
3. Hugo A. Badillo
NSA
Fort Meade, MD
4. George Bieber
OSD
Washington, DC
5. RADM Joseph Burns
Fort George Meade, MD
6. John Campbell
National Security Agency
Fort Meade, MD
7. Deborah Cooper
DC Associates, LLC
Roslyn, VA
8. CDR Daniel L. Currie
PMW 161
San Diego, CA
9. Louise Davidson
National Geospatial Agency
Bethesda, MD
10. Vincent J. DiMaria
National Security Agency
Fort Meade, MD
11. LCDR James Downey
NAVSEA
Washington, DC

12. Dr. Diana Gant
National Science Foundation
Arlington, VA
13. Jennifer Guild
SPAWAR
Charleston, SC
14. Richard Hale
DISA
Falls Church, VA
15. LCDR Scott D. Heller
SPAWAR
San Diego, CA
16. Wiley Jones
OSD
Washington, DC
17. Russell Jones
N641
Arlington, VA
18. David Ladd
Microsoft Corporation
Redmond, WA
19. Dr. Carl Landwehr
National Science Foundation
Arlington, VA
20. Steve LaFountain
NSA
Fort Meade, MD
21. Dr. Greg Larson
IDA
Alexandria, VA
22. Penny Lehtola
NSA
Fort Meade, MD

23. Ernest Lucier
Federal Aviation Administration
Washington, DC
24. CAPT Deborah McGhee
Headquarters U.S. Navy
Arlington, VA
25. Dr. Vic Maconachy
NSA
Fort Meade, MD
26. Doug Maughan
Department of Homeland Security
Washington, DC
27. Dr. John Monastra
Aerospace Corporation
Chantilly, VA
28. John Mildner
SPAWAR
Charleston, SC
29. Jim Roberts
Central Intelligence Agency
Reston, VA
30. Charles Sherupski
Sherassoc
Round Hill, VA
31. Dr. Ralph Wachter
ONR
Arlington, VA
32. David Wirth
N641
Arlington, VA
33. Daniel Wolf
NSA
Fort Meade, MD

34. Jim Yerovi
NRO
Chantilly, VA
35. CAPT Robert Zellmann
CNO Staff N614
Arlington, VA
36. Dr. Cynthia E. Irvine
Naval Postgraduate School
Monterey, CA
37. Thuy D. Nguyen
Naval Postgraduate School
Monterey, CA
38. CAPT Deborah McGhee
Naval Postgraduate School
Monterey, CA
39. Lily Tse
Naval Postgraduate School
Monterey, CA